

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Aleš Šavli

**RAZVOJ MULTIMEDIJSKEGA UČNEGA  
OKOLJA**

DIPLOMSKO DELO  
NA VISOKOŠOLSKEM STROKOVNEM ŠTUDIJU

Mentor: viš. pred. dr. Igor Rožanc

Ljubljana, 2010



Št. naloge: 00480/2009

Datum: 15.10.2009

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **ALEŠ ŠAVLI**

Naslov: **RAZVOJ MULTIMEDIJSKEGA UČNEGA OKOLJA**  
**DEVELOPMENT OF MULTIMEDIA LEARNING ENVIRONMENT**

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija

Tematika naloge:

Multimedijska učna okolja služijo za sistematičen prikaz večpredstavnostnih učnih vsebin učencem, ki se na interaktiven način učinkoviteje učijo izbrano snov. Njihova prednost je uporaba različnih načinov posredovanja snovi, s čemer učenje postane privlačnejše.

V diplomski nalogi najprej opredelite pojem učnega okolja, nato pa izberite in kratko predstavite nabor primernih obstoječih tehnologij in orodij, iz katerih sestavite svoje multimedijско učno okolje. Pri razvoju od idejne zasnove do izdelanega okolja uporabite agilni metodološki pristop, težišče naloge pa naj bo na prikazu programske izvedbe povezave posameznih delov sistema, ki so izvedeni z različnimi orodji. Izgled učnega okolja prikažite na primeru poučevanja angleščine na osnovnošolskem nivoju.

Mentor:

viš. pred. dr. Igor Rožanc



Dekan:

prof. dr. Franc Solina

Rezultati diplomskega dela so intelektualna lastnina Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavljane ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje Fakultete za računalništvo in informatiko ter mentorja.

**Namesto te strani vstavite original izdane teme diplomskega dela s podpisom mentorja in dekana ter žigom fakultete, ki ga diplomant dvigne v študentskem referatu, preden odda zdelek v vezavo!**



# **IZJAVA O AVTORSTVU**

## **diplomskega dela**

Spodaj podpisani/-a Aleš Šavli,

z vpisno številko 63030056,

sem avtor/-ica diplomskega dela z naslovom:

Razvoj multimedijskega učnega okolja

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal/-a samostojno pod mentorstvom (naziv, ime in priimek)  
viš. pred. dr. Igor Rožanc
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.)  
ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki »Dela FRI«.

V Ljubljani, dne 14.04.2010

Podpis avtorja/-ice: \_\_\_\_\_



## **Zahvala**

Zahvaljujem se mentorju na Fakulteti za računalništvo in informatiko v Ljubljani viš. pred. dr. Igorju Rožancu in učiteljici Viljenki Šavli za vso pomoč in podporo pri izdelavi diplomske naloge.

Zahvaljujem se staršem, ki so mi omogočili študij in me podpirali, ter vsem ostalim, ki so kakorkoli prispevali k nastanku te diplomske naloge.





# KAZALO

POVZETEK .....	1
ABSTRACT .....	2
1. UVOD.....	3
2. UPORABLJENE TEHNOLOGIJE IN MULTIMEDJSKA ORODJA.....	5
2.1 Tehnologije učnega okolja.....	7
2.1.1 Ogrodje Microsoft .NET .....	7
2.1.2 Microsoft Visual Studio 2008 .....	8
2.1.3 Programski jezik C# .....	9
2.1.4 Adobe Flash CS3 .....	9
2.2 Multimedjska učna orodja .....	10
2.2.1 Quizfaber .....	10
2.2.2 jClic .....	11
2.2.3 iSPRING .....	12
2.2.4 Content creator .....	13
2.2.5 Articulate – Engage .....	13
2.2.6 Spellmaster .....	14
2.2.7 Moodle.....	15
2.3 Izbira metodologije programiranja .....	16
3. UČNO OKOLJE.....	18
3.1 Zamisel izdelave .....	18
3.2 Opis rešitve .....	22
3.2.1 Podatkovni model.....	22
3.2.2 Izdelava aplikacije .....	24
3.2.2.1 Uporabniški vmesniki .....	24
3.2.3 Podatkovna struktura .....	34
3.2.4 Praktični zgled .....	37
4. ZAKLJUČKI .....	38
5. VIRI.....	39

## **Seznam uporabljenih kratic in simbolov**

CIL – vmesna koda (angl. Common Intermediate Language)

CLI – skupna jezikovna infrastruktura (angl. Common Language Infrastructure)

CLR – skupni jezikovni izvajalnik (angl. Common Language Runtime)

e-gradivo – elektronsko gradivo

e-slovar – elektronski slovar

IDE – integrirano razvojno okolje (angl. Integrated Development Environment)

i-delovni zvezek – interaktiven delovni zvezek

i-učbenik – interaktiven učbenik

JIT – prevajanje ravno v pravem času (angl. Just in time Compiling)

## POVZETEK

Diplomska naloga opiše ideje in razvoj multimedijskega učnega okolja v povezavi z uporabo dostopnih razvojnih tehnologij in multimedijskih orodij za izdelavo učnih vsebin. Dotakne se pojma učnega okolja, oriše zgodovino uporabe računalniške opreme v šolah, omeni probleme pri njeni uporabi in utemelji izbiro metodologije programiranja.

V uvodu je predstavljena zgodovina uporabe informacijsko komunikacijske tehnologije v izobraževanju, opisane so različne vrste elektronskih učnih gradiv ter težave in pomankljivosti pri njihovi uporabi. Ugotovljena dejstva in opažene težave pri uporabi trenutno aktualnih elektronskih učnih okolij so bila motiv za razvoj novega multimedijskega učnega okolja, predstavljenega v diplomski nalogi.

Sledi predstavitev klasifikacije multimedijske programske opreme v skupine in utemeljitev izbire primerne systemske platforme. Izbrana tehnologija za razvoj okolja in multimedijska orodja za izdelavo učnih vsebin s prevladujočo tehnologijo Adobe Flash omogočajo napredno upravljanje z zvokom, videom in animacijami. Podrobneje je opisna uporaba orodij za izdelavo interaktivnih kvizov (Orodje Quizfaber), za izdelavo niza različnih interaktivnih nalog (jClic), za inovativno pretvorbo PowerPoint predstavitev v tehnologijo Flash in izdelavo kvizov (iSPRING), za izdelavo didaktičnih iger in iger za učenje besed (Content creator, Spellmaster) in za izdelavo animiranega slovarja (Articulate – Engage). Predstavljeno pa je tudi v slovenskih šolah precej razširjeno e-učno okolje Moodle.

Osrednji del diplomske naloge najprej utemeljuje izbiro metodologije programiranja za izbrano učno okolje. Zaradi prilagodljivejšega sodelovanja med izvajalcem in naročnikom je bila izbrana agilna metodologija programiranja. Opisane so značilnosti, prednosti in slabosti te tehnologije. Pojem učnega okolja je razdeljen v tri skupine: formalno, neformalno in priložnostno. Zasnovano multimedijsko učno okolje se deli na več vsebinskih sklopov: i-učbenik in i-delovni zvezek, interaktivne naloge, pesmice in zgodbe, igre in e-slovarček. I-učbenik in i-delovni zvezek sta elektronska predstavitev klasičnega učbenika in delovnega zvezka z dodatnimi interaktivnimi kvizi. Interaktivne naloge omogočijo utrjevanje, ponavljanje in preverjanje znanja s pomočjo različnih tipov nalog. Animirane pesmice in zgodbe so didaktičen pripomoček za popestritev vsebine. Igre, izdelane v tehnologiji Flash, omogočajo dodatno zanimivejše utrjevanje snovi. Slovar daje možnost sprotnega vpogleda v pomen besed.

Sledi podrobnejši opis primera rešitve izdelave multimedijskega učnega okolja. Najprej je predstavljen podatkovni model in prikazana sestava uporabniških vmesnikov, ki omogočajo izbiro med posameznimi vsebinskimi sklopi, med ponujenimi nalogami, možne prehode med njimi ter prikaz učnih vsebin. Prikaz vsebine posameznega sklopa je izdelan z uporabo uporabniških kontrolnikov (angl. User control), prikaz učnih vsebin pa je izveden z uporabo komponent WebBrowser in axShockwaveFlash. Podrobno so opisane podatkovne strukture, kjer so shranjene vse informacije o učnih vsebnah in razred, ki se uporablja za grafično obdelavo gumbov. Razred ima definirane funkcije za dodajanje napisov na gumba. Ti omogočajo izbiro med posameznimi nalogami. Za konec je ilustrativno prikazan praktični izgled uporabe učnega okolja.

### Ključne besede:

multimedijsko učno okolje, interaktivna gradiva, agilna metodologija, multimedijska razvojna orodja, ogrodje .NET, uporabniški vmesnik, Adobe Flash

## ABSTRACT

The thesis describes ideas and development of multimedia learning environment in connection with available development technologies and multimedia tools for producing learning objects. It also refers to the concept of learning environment, outlines the history of the use of computer technology in schools and mentions problems in usage and establishes the choice of programming methodology.

In introduction the history of Information and communication technologies usage in education is presented and different types of e-learning objects, problems or deficiencies in usage are described. The stated facts and problems noted in the usage of e-learning environments were the motives for the development of a new multimedia learning environment presented in my thesis.

The classification of multimedia programming equipment and argumentation for the chosen system platform is described next. Both chosen technology for developing environments and multimedia tools include Adobe Flash technology so they enable advanced administration and management of voice, video and animation. Tools for making series of interactive quizzes (Quizfaber), series of interactive exercises, (jClic), innovative transformation of PowerPoint presentations into flash technology and quizzes (iSpring), didactic games and games for learning words (Content Creator, Spellmaster) and animated dictionary (Articulate – Engage) are described in details. There is also a presentation of Moodle, an often used e-learning environment in Slovenian schools.

In the main part the thesis first arguments why the programming methodology for the learning environment was chosen. Because of the need for more adaptable cooperation between a developer and a customer the agile programming methodology was chosen and the advantages and disadvantages of it are described. The concept of learning environment is divided into three groups: formal, informal and occasional. The designed multimedia learning environment is divided into different content clusters: i-course book, i-workbook, i-exercises, songs and stories, games and a dictionary. I-course book and i-workbook are electronic versions of the classic course book and workbook with some interactive quizzes added. Different types of i-exercises enable students to revise and check their knowledge. The animated songs and stories are additional didactic resources that make learning more visual. The games produced in Flash offer interesting and playful learning. Interactive dictionary gives a simultaneous insight into the meaning of the words.

After that a solution of the new multimedia environment is described in details. The data model is presented and also the structure of the user interfaces that enable the choice, the shift between several content clusters and exercises and all the learning content. The content cluster view was created by the user control the learning content view was made by the WebBrowser component and axShockwaveFlash. The same way data structures and information about learning objects are described in details. The class includes defined functions for adding names on the buttons that offer choices between different exercises. The thesis finally presents an illustrative overview of how the learning environment is used.

### Keywords:

multimedia learning environment, interactive learning materials, agile methodology, multimedia development tools, .NET Framework, user interface, Adobe Flash

## 1. UVOD

V 90-tih letih se je pri nas začel strm vzpon uporabe računalnikov v izobraževanju. Učitelji so imeli na voljo CD-rome z animiranimi zgodbicami, pesmicami, učnim gradivom, opremljenim z grafiko in hiperpovezavami, enciklopedije itd. Z razvojem tehnologije in njeno množičnejšo uporabo, je le-ta postajala vse cenejša, zato je večina šol, s podporo ministrstva za šolstvo, izkoristila možnost in ustrezno opremila računalniške učilnice, kjer so se učenci lahko srečali z računalnikom in spoznali večplastno uporabnost te tehnologije. V nekaterih šolah so za posamezne predmete celo uredili nekakšne multimedijske učilnice, kjer je učitelj upravljal z računalnikom, učenci pa so preko monitorjev spremljali animacije, video predstavitve oziroma zvočno in grafično obogateno gradivo, ki jim ga je učitelj prikazoval. To je bil že velik korak naprej od klasičnega učenja, kjer se uporablja le tablo in kreda ter učbenik. A ta način učenja je bil še vedno enosmeren in ni omogočal neposredne interaktivnosti.

S pocenitvijo računalnikov, prihoda širokopasovnega dostopa do interneta in novimi spletnimi tehnologijami, se je močno razširila uporaba računalnikov v šolah na vseh stopnjah. Država je šolam preko razpisov omogočala nakupe računalnikov po ugodnih cenah in kmalu so bile šole polne računalnikov. Tu pa je nastopila nova težava, saj zaradi pomanjkanja ustreznih in dobrih didaktičnih gradiv, zmogljivosti računalnikov v šoli ni bilo mogoče učinkovito izkoristiti. Računalnike se je namreč na začetku uporabljalo skoraj izključno za učenje, kako rokujemo z računalnikom oziroma, kako ga uporabljamo. Uporaba računalnika za druge namene, to je pri drugih šolskih predmetih, je bila večino omejena na pisanje seminarskih nalog, predstavitev, ali oblikovanje gradiv za plakate. Uporaba računalnikov pri tujih jezikih, matematiki, biologiji ali slovenščini je bila po večini zelo omejena, saj učitelji niso imeli na voljo ustreznega elektronskega gradiva, ki bi jim to omogočalo. Nekateri učitelji so se znašli in doma v prostem času sestavljali zanimive predstavitve, ki so jih nato učencem v šoli prikazovali preko projektorjev.

Danes je razvoju računalniške tehnologije praktično nemogoče slediti. Priča smo razcvetu uporabe spletnih tehnologij, ki omogočajo različne možnosti za uporabo v izobraževalnem procesu v šolah in pri učenju doma. Za učinkovit pretok gradiv in informacij med učitelji in učenci pa je ob tehnologiji potrebno imeti tudi nekoga, ki omogoča, da učenje postane živo, dinamično ter vzpodbuja interaktivnost. V ta namen ministrstvo za šolstvo in šport pripravlja razpise na temo usposabljanja učiteljev za uporabo informacijsko komunikacijske tehnologije (angl. Information and communication technologies) pri poučevanju in učenju. V letih 2007 in 2008 je ministrstvo za šolstvo in šport preko evropskega socialnega sklada izdalo razpis za izdelavo elektronskih učnih gradiv, ki se jih dobi na spletnem naslovu [9]. Gradiva morajo biti skladna s standardom SCORM/AICC [2], kar pomeni, da se jih lahko pretoči in neposredno vključi kot učno gradivo v spletnem učnem okolju, kot je na primer spletna učilnica Moodle [24].

Ministrstvo v zadnjih letih vzpodbuja tudi uporabo novih interaktivnih tabel kot učnih pripomočkov pri poučevanju in učenju na šolah. Interaktivne table (angl. Interactive

whiteboard) so nova tehnologija. V principu gre za povezavo računalnika in table, saj se namizje računalnika preko projektorja projecira na tablo. Z dotikom table s prstom ali posebnim pisalom upravljamo računalnik brez uporabe miške ter tipkovnice. (slika 1)

Interaktivna tabla predstavlja nekakšen velik ekran na dotik, prikaz na njem pa omogoča projektor. Tehnologija dotika se od proizvajalca do proizvajalca spreminja, pri nekaterih tablah je možna uporaba le s posebnimi pisali [28], drugi pa omogočajo uporabo z dotikom prsta ali pisal [29].



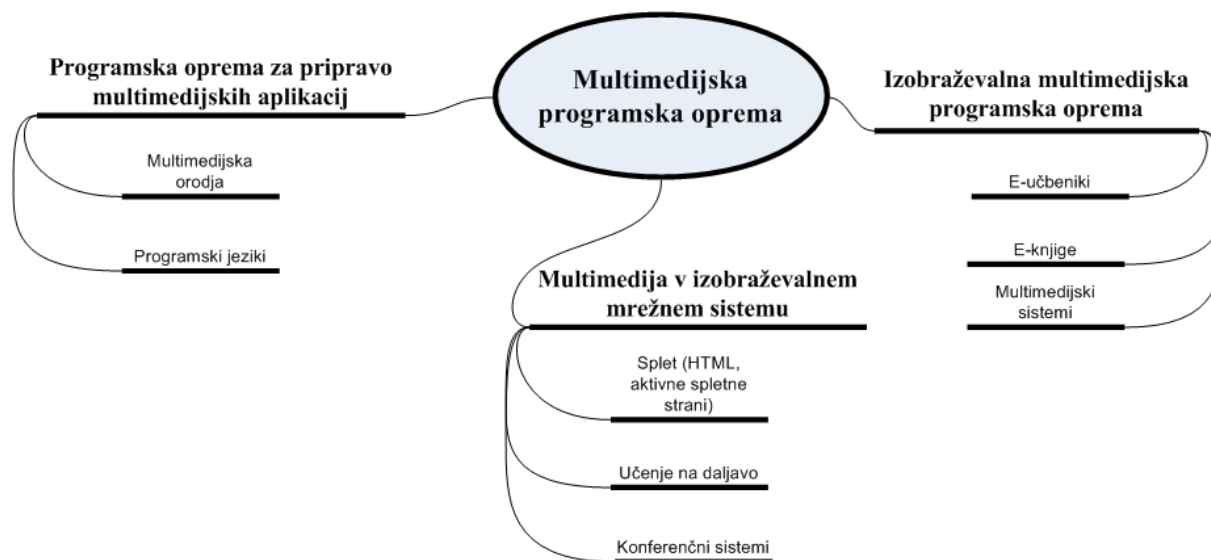
**Slika 1: Primer interaktivne table**

Takšna tabla lahko služi kot nadomestek za šolske table, ki poleg tradicionalne rabe omogoča tudi prikaz vseh stvari, ki jih lahko predstavimo na računalniškem namizju (izobraževalni računalniški programi, spletne strani, aplikacije itd.). Priložena programska oprema omogoča učiteljem, da izdelajo svoje gradivo in da posnamejo uporabo gradiva ali predstavitev z učenci in jih shranijo za kasnejši ogled. Ta možnost omogoča učencem, da si določeno snov lahko ponovno ogledajo in jo tako ponovijo in utrdijo. Njihova didaktična vrednost je v veliki meri odvisna od tega, kako uspešno in smiselno jih učitelj vključuje v pouk.

Upoštevajoč vsa omenjena dejstva in težave prihajamo do ugotovitve, da za učinkovito rabo interaktivnih tabel potrebujejo učitelji ustrezno multimedijsko učno okolje, ki jim bo pomagalo vključevati zanimive interaktivne učne vsebine neposredno v pouk. V današnjem času, ko je informacijsko komunikacijska tehnologija takorekoč prisotna na vsakem koraku in ureja že doberšen del našega vsakdanjega življenja, se lahko takšne učne vsebine in multimedijsko učno okolje izdelata relativno enostavno. Za realizacijo omenjenega sta potrebna dva ključna elementa: uporabne in tematsko primerne interaktivne učne vsebine ter učno okolje, ki bo te vsebine uporabniku prijazno, sistematično in učinkovito prikazovalo. Če želimo izdelati takšne učne vsebine, je potrebno imeti primerna multimedijska orodja, s katerimi lahko izdelamo željene učne vsebine. Na spletu je takšnih multimedijskih orodij veliko. Nekatera orodja so brezplačna, ostala pa lahko v večini primerov brezplačno preizkusimo. Pri izdelavi multimedijskega učnega okolja pa lahko uporabimo znane tehnologije za razvoj programske opreme. V naslednjih korakih smo dali poudarek razvoju multimedijskega učnega okolja in predstavitvi orodij za izdelavo učnih vsebin.

## 2. UPORABLJENE TEHNOLOGIJE IN MULTIMEDIJSKA ORODJA

Dr. Ivan Gerlič je [8, 14] multimedijško programsko opremo v izobraževanju razdelil na tri glavne skupine (slika 2): programska oprema za pripravo multimedijskih aplikacij, izobraževalna multimedijška programska oprema in multimedija v izobraževalnem mrežnem sistemu.



Slika 2: Primer delitve multimedijske programske opreme

V skupino programske opreme za pripravo multimedijskih aplikacij spadajo multimedijška orodja s katerimi izdelamo interaktivne učne vsebine, obdelujemo slike, zvok, animacije (Adobe Photoshop, Flash, JCLIC, iSpring, Quizfaber ...) in splošni programski jeziki (C#, Java, VB.NET ...). V drugo skupino spada mrežna oprema, ki omogoča učenje preko spleta oziroma omrežja. Uporaba te programske opreme se je v zadnjih letih najbolj razvijala in je trenutno najbolj razširjen način poučevanja. V zadnjo skupino pa spada programska oprema, ki jo predstavljajo elektronske knjige, elektronski učbeniki in elektronski slovarji. Programsko opremo se lahko namesti na lokalnem računalniku. Multimedijško učno okolje, za uporabo na interaktivni tabli lahko razvrstimo v vse tri skupine.

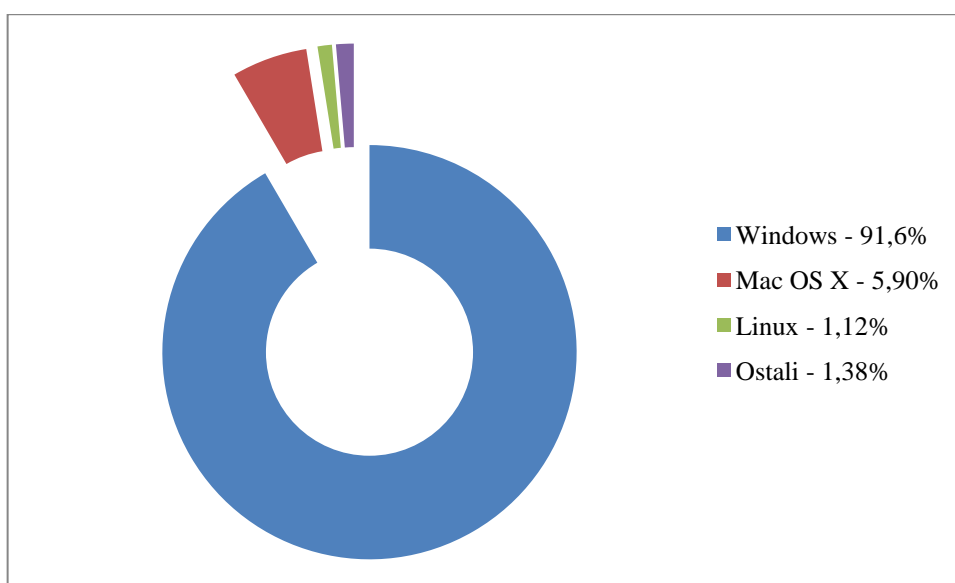
Ko izbiramo tehnologijo izdelave takega gradiva, je potrebno najprej opredeliti način, kako bo aplikacija uporabljena. Danes je večino spletnih aplikacij narejenih s pomočjo spletnih tehnologij (asp.net, php, ruby ...), ki potrebujejo dostop do spleta. V kolikor želimo učno okolje približati čim večjemu številu učencev, je potrebno izdelati okolje, ki ga lahko učenec uporablja tako v šoli kot tudi doma brez spletne povezave. Ena od možnosti je spletna aplikacija, ki se vsakemu učencu namesti na njegov računalnik. V takem primeru bi bilo potrebno namestiti še spletni strežnik in podatkovno bazo. Pri tem pa lahko prihaja do težav s povezavo v paket, ki naj bi bil uporabniku prijazen. Na podlagi izkušenj smo se odločili za uporabo tehnologije windows forms. Rezultat te tehnologije je aplikacija, ki uporablja grafično in sistemsko okolje Microsoft Windows [30] za prikaz uporabniških vmesnikov. Učno okolje se lahko sestavi v namestitveni program, ki ga je mogoče poljubno prenašati med računalniki in omogoča enostavno namestitev učnega okolja na učenčevem računalniku. Za



delovanje je potrebno namestiti ogrodje .NET, ki pa je običajno že nameščeno v operacijski sistem Microsoft Windows. V primeru, da okolje še ni nameščeno, za to poskrbi namestitveni program sam.

Slabost ogrodja .NET je nekompatibilnost z ostalimi operacijskimi sistemi, saj je omejeno na operacijski sistem Microsoft Windows. Za distribucije Linux sicer obstaja odprtokodni projekt MONO, ki pa omogoča omejeno uporabo aplikacij napisanih v ogrodju .NET.

Raziskava o uporabi operacijskih sistemov, opravljena na spletni enciklopediji wikipedia v začetku leta 2010 [26], je pokazala (slika 3), da se v 91,6% primerih uporablja operacijski sistem Windows. Primerjali so dostope do nekaterih spletnih strani, kjer so beležili s katerim operacijskim sistemom je uporabnik prišel na spletno stran. Tudi večina šol v Sloveniji uporablja Microsoftove izdelke, zato razvijanje programske opreme v ogrodju .NET ne predstavlja omejitve pri njegovi uporabi.



Slika 3: Delež uporabe operacijskih sistemov (Februar 2010)

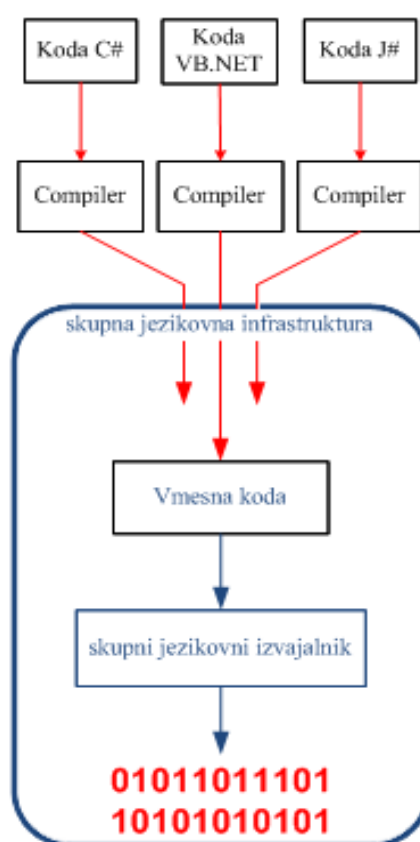
V nadaljevanju sledi opis tehnologije učnega okolja in multimedijskih orodij, ki smo jih uporabili pri izdelavi multimedijskega učnega okolja.

## 2.1 Tehnologije učnega okolja

### 2.1.1 Ogradje Microsoft .NET

Microsoft .NET [11] je razvojno ogrodje za razvoj programske opreme. Uporabljamo ga lahko na operacijskih sistemih Microsoft Windows. S pomočjo ogrodja .NET lahko izdelamo spletne, namizne, aplikacije za pametne naprave in še veliko drugih vrst aplikacij.

Ogradje .NET (slika 4) je sestavljeno iz dveh glavnih komponent: skupnega jezikovnega izvajalnika (angl. Common Language Runtime - CLR) in knjižnice razredov ogrodja .NET (angl. .NET Framework Class Library - FCL). CLR je Microsoftova izvedba koncepta skupne jezikovne infrastrukture (angl. Common Language Infrastructure - CLI), katere namen je ponuditi v različnih jezikih okolje za izdelavo in izvajanje aplikacij, nadzor nad napakami, varnost itd.



Slika 4: Koncept .NET

CLR je temelj ogrodja .NET. Predstavljamo si ga lahko kot navidezni računalnik, ki z uporabo metode »prevajanje ravno v pravem času« (angl. Just in Time Compiling - JIT) vmesno kodo (angl. Common Intermediate Language - CIL) pretvori v strojno kodo. Vmesna koda se generira vsakič pri prevajanju programa. Prevedena koda sama po sebi ni prilagojena za strojno opremo (CPU), zato ogrodje .NET uporablja navidezni računalnik (CLR). Tako dobimo popolno prenosljivost kode, ne glede na strojno opremo, ki jo uporabljamo.

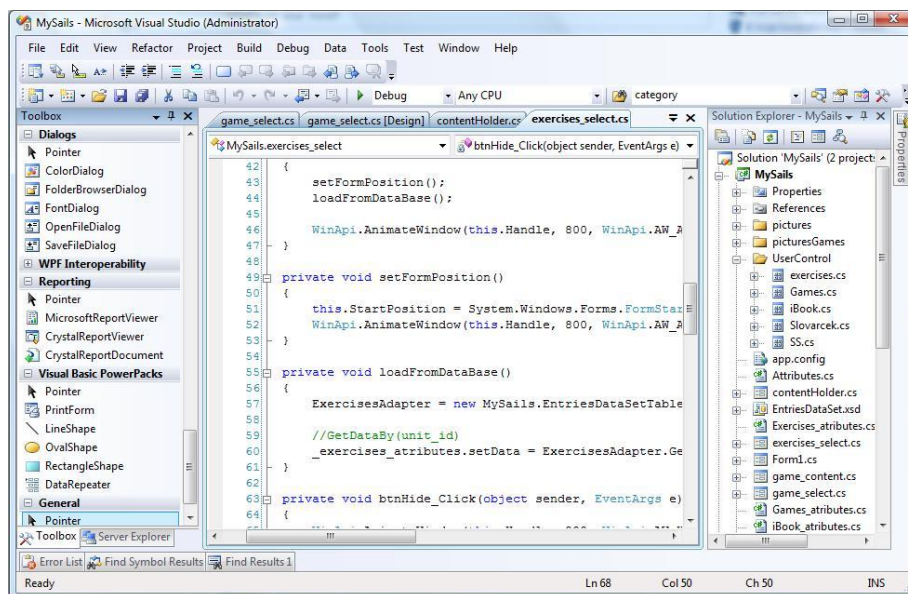
Namen uporabe tehnologije .NET je v izdelavi osrednjega dela multimedijskega učnega okolja – programa, ki ima nadzor nad vsemi učnimi vsebinami.

### 2.1.2 Microsoft Visual Studio 2008

Microsoft Visual Studio [12] je integrirano razvojno okolje (angl. Integrated Development Environment - IDE). IDE (slika 5) je okolje za razvoj aplikacij, ki omogoča programskim razvijalcem lažje delo.

Glavni deli okolja so:

- urejevalnik besedila (programske kode)
- prevajalnik in/ali interpreter
- testiranje in razhroščevanje
- pomoč pri pisanju programske kode (angl. Autocompletion)



Slika 5: Vmesnik Visual Studio 2008

Glavne značilnosti:

- podpora programskim jezikom
  - C/C++, Visual Basic, Visual C#, J#, F#, Python, Ruby
  - podpira tudi JavaScript, XML, HTML in CSS
- omogoča izdelavo različnih visokokakovostnih programskih rešitev
  - namizne aplikacije
  - spletne aplikacije
  - spletne storitve
  - mobilne aplikacije
- ima orodja za dostop do vseh večjih podatkovnih baz
- orodja za razhroščevanje, konstruiranje, testiranje in razširjanje
- izrablja vse funkcije ogrodja .NET

Microsoft Visual Studio 2008 smo uporabili kot razvojno okolje multimedijskega učnega okolja s pomočjo ogrodja .NET.

### 2.1.3 Programski jezik C#

Programski jezik C# (angl. C-sharp) [5, 27] je moderen objektno orientiran programski jezik, ki je bil zgrajen na temeljih programskih jezikov C, C++ in Java. Zgrajen je bil kot odgovor na programski jezik Java in se uporablja za razvijanje aplikacij v ogrodju .NET [11].

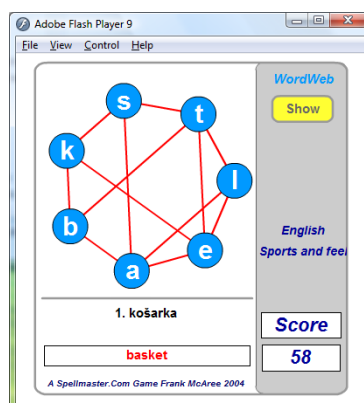
Glavne značilnosti:

- vmesniki (angl. Interfaces)
- strukture (angl. Structs)
- delegati (angl. Delegates)
- lastnosti (angl. Properties)
- dogodki (angl. Events)

Programski jezik smo uporabili pri razvoju ogrodja multimedijskega učnega okolja v povezavi z ogrodjem .NET in razvojnim okoljem Microsoft Visual Studio.

### 2.1.4 Adobe Flash CS3

Večina multimedijskih učnih orodij je izdelana s pomočjo multimedijskega orodja Adobe Flash. Orodje Flash je zelo popularno pri izdelavi spletnih strani. S pomočjo Flash-a se je pričelo na spletne strani enostavno dodajati zanimive animacije in zelo uporabno interaktivnost. Danes se večinoma uporablja za izdelavo animacij, ki jih uporabljajo v oglaševalske namene in izdelavo različnih komponent za predvajanje videa na spletnih straneh (Youtube, Google video, Vimeo,...).



Slika 4: Primer interaktivne učne vsebine izdelane v tehnologiji Flash

Orodje lahko uporabimo tudi za upravljanje z zvokom in videom, lahko pa se tudi povežemo s podatkovno bazo, če želimo izdelovati spletne aplikacije. Pri tem se uporablja skriptni jezik ActionScript (trenutno verzija 3.0), s katerim lahko aplikacijam, animacijam ali videu, dodajamo dodatno funkcionalnost.

Za predvajanje vsebin je potrebno namestiti samostojen Adobe Flash predvajalnik. V spletnih brskalnikih, kjer se večinoma uporablja Flash, je predvajalnik dodan kot vtičnik (angl. Plug-in). V primeru ko predvajalnik ni nameščen, se ga lahko prosto dobi na proizvajalčevi strani [22]. Po nekaterih podatkih je predvajalnik nameščen na kar 96 odstotkih računalnikov, ki so priključeni na medmrežje [1, 10].

## 2.2 Multimedijska učna orodja

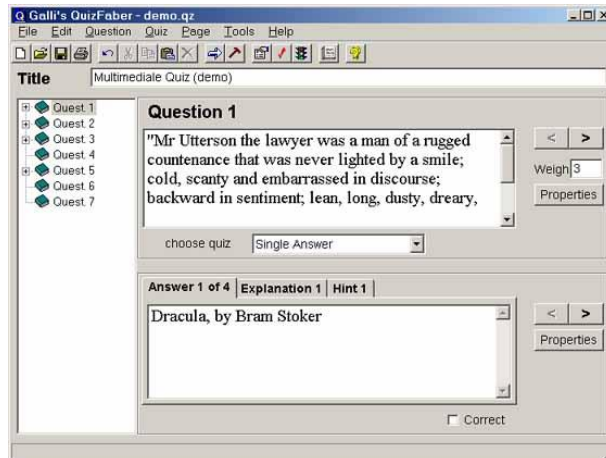
Za izdelavo multimedijskega učnega okolja je potrebno poleg ogrodja izdelati tudi multimedijske učne vsebine. V vsakem učnem okolju je najpomembnejši vsebinski del, ki je temelj vseh učnih okolij, dobra izvedba pa zaokroži učno okolje v zaključeno celoto. Učne vsebine morajo biti vsebinsko bogate in učenca prijazne za uporabo. Na spletu lahko najdemo veliko multimedijskih orodij, ki omogočajo izdelavo vseh vrst učnih vsebin, od brezplačnih pa do zelo dragih.

Glavna tehnologija multimedijskih učnih orodij, ki se jih lahko dobi na spletu, je tehnologija Flash, nekaj orodij pa deluje v tehnologiji Java. V primeru uporabe tehnologije Flash je v večini primerov učna vsebina pripravljena kot spletna stran, ki preko HTML oznak OBJECT prikažejo vsebino. Če spletne strani ne želimo uporabiti, lahko uporabimo samo animacijo flash in jo potem integriramo v svoje okolje.

V nadaljevanju bomo opisali nekatera orodja, ki smo jih uporabili pri izdelavi multimedijskih učnih vsebin.

### 2.2.1 Quizfaber

Quizfaber je prostodostopno orodje [19] za izdelavo spletnih interaktivnih kvizov z uporabo tehnologije JavaScript. Učitelj za izdelovanje kvizov potrebuje grafični urejevalnik kvizov. S pomočjo urejevalnika (slika 5) lahko vnese željena vprašanja, doda slike, zvok itd. Program nato avtomatično izdela spletno stran, ki vsebuje kviz. Učenec lahko rezultate kviza natisne ali jih pošlje po e-pošti.



Slika 5: Urejevalnik kviza

S pomočjo orodja QuizFaber se lahko izdela in ureja veliko tipov vprašanj:

- vprašanja z več odgovori, kjer je le eden pravilen
- vprašanja z več odgovori, eden ali več pravilnih
- vprašanja z odgovori drži in ne drži
- vprašanja z odgovorom, kjer se odgovor prosto vpiše v polje
- vpisovanje manjkajočih besed
- povezovanje besed, ki so vsebinsko povezane

Orodje QuizFaber smo uporabili za izdelavo kviza v vsebinskem sklopu *Exercises*, kjer učenec izvaja različne vaje (izdelane s pomočjo orodja jClic). Ob koncu vseh vaj se lahko preizkusi v omenjenem kvizu.

### 2.2.2 jClic

JClic je odprtokodno orodje za izdelavo niza interaktivnih nalog [18]. Objavljeno je pod licenco GPL (angl. GNU General Public License), tako da se lahko prosto uporablja, razširja in spreminja pod določenimi pogoji. Omogoča izdelavo interaktivnih nalog na lokalnem računalniku z uporabo priloženega predvajalnika ali preko spleta (z javanskim apletom). Celotno orodje je zgrajeno v tehnologiji Java, zato ni omejeno pri uporabi v Windows okolju.

Sestavljen je iz štirih aplikacij:

- **JClic aplet** omogoča vključitev nalog za ogled na spletu.
- **JClic player** je neodvisen program, ki omogoča predvajanje nalog iz lokalnega diska ali lokalnega omrežja brez potrebe po internetni povezavi.
- **JClic author** je glavno orodje za izdelavo, urejanje in objavo nalog v pregledno obliko
- **JClic reports** je sistem zbiranja in generiranje poročil o rezultatih, ki so jih opravili uporabniki



Slika 6: Nekaj primerov nalog

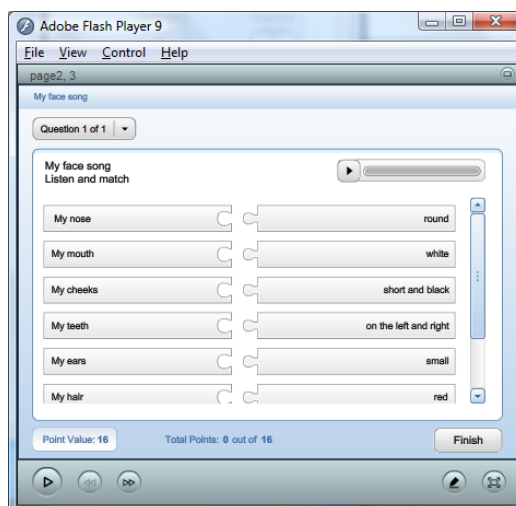
jClic omogoča (slika 6):

- vključevanje grafike, zvoka in videa,
- izbiro grafičnih tem,
- jasno navigacijo,
- zanimive možnosti za izvedbo različnih aktivnosti,
- niz možnosti in izbir tipov nalog,
- pomoč pri reševanju,
- spremljanje števila poskusov, pravilnosti rešitve in popravkov,
- uporabniku prijazen uporabniški vmesnik

Orodje jClıc smo uporabili v vsebinskem sklopu *Exercises*, kjer učenec s pomočjo orodja izvaja različne naloge. V praksi se je orodje izkazalo za zelo enostavno, s katerim se lahko zelo hitro izdelata potrebne učne vsebine.

### 2.2.3 iSPRING

iSPRING je orodje za inovativno pretvorbo predstavitev PowerPoint v Flash tehnologijo z vsemi animacijami in prehodi [17]. Orodje je primerno za spletno objavljjanje, oglaševanje, e-učenje in za ustvarjanje bogato opremljenih medijskih vsebin.



Slika 7: Primer kviza

iSPRING omogoča:

- izdelavo kvizov z različnimi tipi vprašanj (slika 7),
- snemanje in sinhroniziranje videa,
- dodajanje Flash in Youtube video predstavitev,
- dodajanja in snemanje zvočnih posnetkov,
- izbiro med različnimi oblikami predvajalnikov.

Orodje tudi omogoča generiranje SCORM/AICC kompatibilnih kvizov, ki jih nato lahko vključimo ali naložimo v željen sistem upravljanja z učnimi vsebinami (angl. Learning management system – LMS)



Slika 8: Primer uporabniškega vmesnika za izdelavo

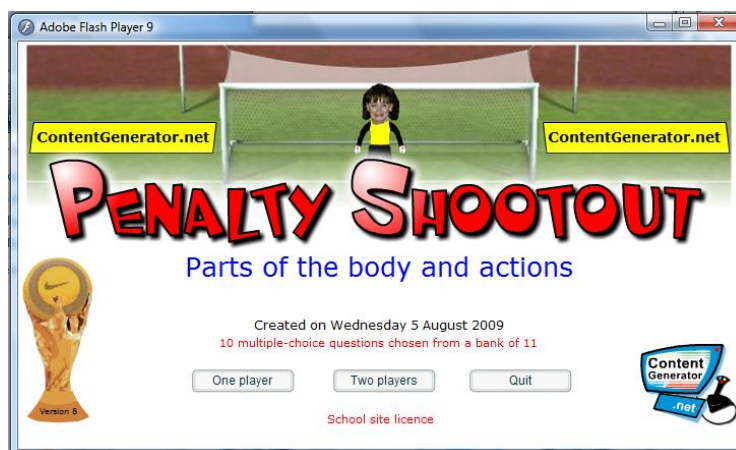
iSPRING se uporablja kot dodatek v programu Microsoft Office Powerpoint. Ko se ga namesti z uporabo namestitvenega programa, se v programu PowerPoint doda nov zavihek (slika 8). Tam so na voljo vse funkcije orodja.



To orodje je glavno orodje celotnega učnega okolja. S pomočjo tega orodja smo izdelali vse učne vsebine v sklopu, kjer je predstavljen elektronski učbenik (iBook) ter pesmice in zgodbe (Songs And Stories).

#### 2.2.4 Content creator

Orodje Content creator omogoča prijazno in enostavno izdelavo interaktivnih multimedijskih vsebin v obliki didaktičnih iger in različnih kvizov, ki ne potrebuje posebnega programiranja [16]. Štirje moduli didaktičnih iger so prostodostopni, ostali pa so licenčni in dobavljivi po dostopni ceni.



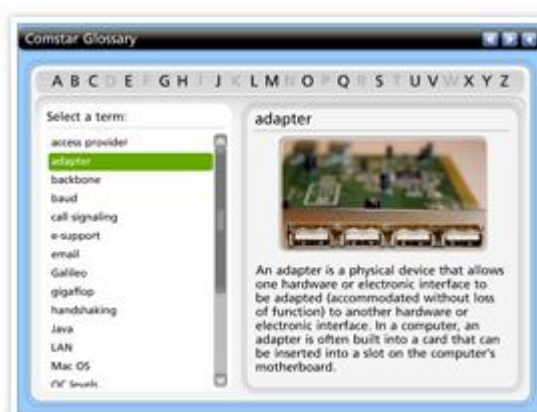
Slika 9: Primer izdelane igre s pomočjo orodja

Gradivo izdelano s pomočjo orodja Content creator je v obliki Flash (slika 9). Predvajamo ga lahko v sklopu spletne strani ali pa samostojno kot animacijo Flash.

V učnem okolju smo orodje uporabili za izdelavo treh iger v vsebinskem sklopu Games.

#### 2.2.5 Articulate – Engage

Orodje Articulate Engage je eno izmed štirih multimedijskih orodij, ki jih ponuja podjetje Articulate [15]. Orodje je plačljivo, omogočeno pa je brezplačno preizkušanje.



Slika 10: Primer animiranega slovarja

Articulate – Engage omogoča izdelavo različnih tipov interakcij (slika 10):



- izdelavo zanimivih grafično animiranih opisov,
- izdelavo animiranega krožnega diagrama z opisi,
- izdelavo piramidnega diagrama z opisi,
- izdelava interaktivne časovnice,
- izdelava zanimivega vmesnika z pogostimi vprašanji (angl. Frequently asked questions - FAQ) in
- izdelavo animiranega slovarja

Tudi gradivo izdelano s pomočjo programov Articulate, je v obliki Flash. Uporabili smo ga za izdelavo slovarja (slika 10) v vsebinskem sklopu Dictionary, za pomoč pri učenju ob uporabi celotnega multimedijskega učnega gradiva. Gradivo je izdelano kot samostojen program in za prikaz ne potrebuje spletnega brskalnika ali kakšnega drugega okolja.

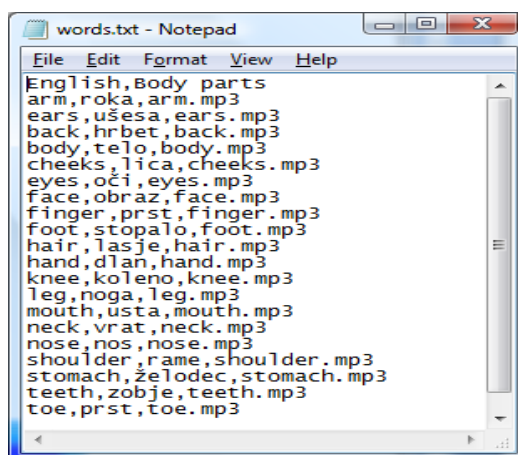
### 2.2.6 Spellmaster

Spellmaster je prostodostopno orodje za pripravo igrer za učenje besed v tehnologiji Flash [20]. Del je tudi zvočno opremljen.



Slika 11: Primer igre z poslušanjem

Orodje je zelo enostavno, saj ponuja že vnaprej izdelane grafične predloge, s katerimi nato upravljamo. Z igro se učenec uči besede tako, da jih povezuje na podlagi branja slovenskega prevoda ali poslušanja zvočnega posnetka. Ko učenec poveže vsa prosta polja, mu igra prikaže dosežen rezultat.



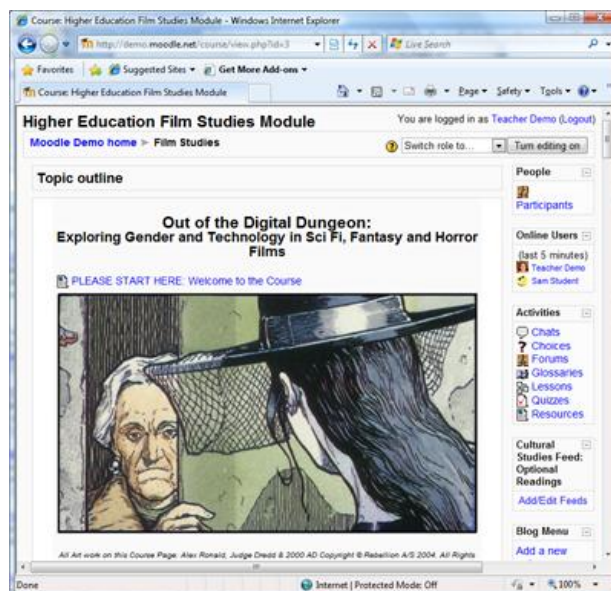
Slika 12: Primer tekstovne datoteke

Igra deluje tako, da preko zunanje tekstovne datoteke (slika 12) prebere vse zahtevane besede, ki jih nato prikaže. Uporabili smo jo za izdelavo štirih iger v vsebinskem sklopu Games.

### 2.2.7 Moodle

Spletno okolje Moodle [24] je zelo razširjeno učno okolje v šolah ter spada med sisteme za upravljanje z vsebinami (angl. Content management system - CMS). Omogoča pripravo ter izvedbo izobraževanj, ki se izvajajo na daljavo. Beseda Moodle ima dva pomena:

1. Kratica za modularno objektno usmerjeno dinamično okolje za poučevanje (angl. **Modular Object-Oriented Dynamic Learning Environment**)
2. Angleški izraz za »ukvarjati se s čim sproščeno in brez posebnega načrta«



Slika 13: Primer Moodla

Moodle (slika 13) omogoča enostaven dostop in upravljanje okolja za pripravo gradiv, uporabo teh gradiv ter spremljanje njihove uporabe. Je spletna aplikacija, zato je dostop do nje omogočen z uporabo spletnega brskalnika brez namestitve dodatnih aplikacij.

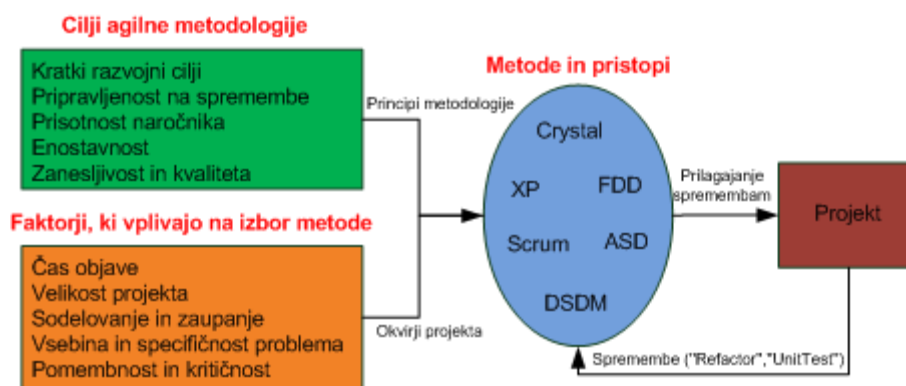
Šole lahko preko akademske in raziskovalne mreže Slovenije – ARNES pridobijo spletno mesto, kamor tak sistem namestijo, lahko pa ga namestijo na lokalni šolski strežnik in uporabljajo v lokalnem omrežju. Sistem je razvit v spletnem programskem jeziku PHP v povezavi z različnimi podatkovnimi bazami, tako da ga lahko namestimo na večino operacijskih sistemov (Windows, Linux, Mac OS).

## 2.3 Izbira metodologije programiranja

Razvoj multimedijskega učnega okolja zahteva veliko sodelovanja med izvajalcem in naročnikom, v našem primeru med programerjem in učiteljem. Iz tega stališča je agilna metodologija primerna izbira.

Pri agilni metodologiji (slika 14) je sodelovanje med izvajalcem in naročnikom tesno povezano. Programsko kodo se večkrat popravlja, pri čemer je potrebno vedno zagotoviti ustrezno testiranje kode. Metodologija [3, 4] aktivno vključuje naročnike, saj naročniku daje možnost pogostih sprememb ali novih zahtev. Primerna je za projekte s kratkim razvojnem ciklom, kjer je potrebno hitro postavljanje delnih rešitev v produkcijo ter tesno sodelovanje. Metodologija ima v primerjavi s klasičnimi plansko vodenimi metodologijami manjši poudarek na poslovnih procesih in dokumentaciji. To ne pomeni, da načrtovanja ni, ampak je načrtovanje fleksibilnejše, oziroma se izvaja na osnovi trenutnega znanja o projektu.

Za agilno metodologijo je značilno, da je rastoča, prilagodljiva, kooperativna in enostavna. Rastoča, ker se vedno razvija v kratkih razvojnih ciklih. Prilagodljiva, saj omogoča izvedbo sprememb v kratkem razvojnem ciklu in enostavna za učenje, uporabo ter dokumentiranje. Za tesno sodelovanje med naročnikom in izvajalcem je potrebno kooperativno sodelovanje pri razvoju.



Slika 14: Ključni koncepti metodologije

Ena izmed najbolj znanih agilnih metodologij, po kateri smo razvijali aplikacijo, je ekstremno programiranje (angl. Extreme Programming – XP). XP predvideva programiranje v parih, običajen delovni čas (8 ur na dan), skupno lastništvo kode, razvijanje po enostavnih specifikacijah, hiter razvojni cikel itd. Področja dela lahko razdelimo na področje planiranja, načrtovanja, testiranja in kodiranja. Vsako od področij določa specifične postopke. Planiranje določa, da se morajo programerji dobro zavedati poslovne logike, seznanjeni morajo biti s specifikacijami in roki oddaje ter na podlagi teh planirati izdelavo. Med izdelavo so

pomembni tudi dnevni sestanki. Da bi načrtovalci hitreje in učinkoviteje načrtovali aplikacije, so se s časoma razvili vzorci načrtovanja (angl. Design by Patterns), ki predstavljajo jedro rešitev za pogosto nastajajoče težave. Pri načrtovanju je potrebno poleg kodiranja in testiranja kodo tudi popravljati in dograjevati na podlagi danih vzorcev. Med razvojem aplikacije je zelo pomembno testiranje. Pri XP velja princip, da malo testiranja odstrani malo napak, veliko testiranja pa odpravi veliko več napak. Za vsako funkcijo oziroma sklop funkcij je potrebno preveriti ali pravilno deluje. Programer (ali skupina programerjev) tako izdela vse testne primere, ki se jih lahko domisli in bi lahko pokazali slabosti oziroma napake pri izdelavi oziroma planiranju. Ko so vsi testi opravljeni brez napak, je koda delujoča in se lahko nadaljuje razvoj naslednjega sklopa. Pri kodiranju aplikacij vsi programerji sodelujejo med seboj, poznajo kodo drugih programerjev in lahko prevzamejo njihovo delo. Pred posameznim dodajanjem kode mora programer izvesti testiranje, optimizacijo aplikacije pa se izvaja ob koncu projekta. Delovni čas naj ne bi presegal 8 ur na dan.

Agilnim metodologijam pripisujejo tudi nekaj kritik:

- metodologija je tako učinkovita kot so učinkoviti ljudje, razvijalci,
- pomanjkanje dokumentacije,
- deluje le z izkušenimi programerji,
- potrebno veliko število sestankov in
- težko je oceniti, koliko dela je potrebno za izdelavo določenih stvari, ker v prvih fazah razvoja primankuje definiranih zahtev in dokumentacije.

### 3. UČNO OKOLJE

V splošnem lahko učna okolja razdelimo glede na način učenja [23]: formalno (učenje v razredu), neformalno (učenje izven formalnega izobraževalnega sistema) in priložnostno (učenje v vsakdanjiku) učno okolje.

Formalno učno okolje (angl. Formal learning) je vsako okolje, kjer poteka učenje po nekem učnem načrtu. Zagotavlja ga šolska institucija in je skrbno načrtovano (časovno, specifični cilji, pristopi ...). Rezultat takšnega učenja je običajno javno veljavna kvalifikacija in pridobitev spričevala. Izobraževanje v slovenskem učnem sistemu lahko razdelimo na: predšolsko vzgojo, osnovno šolo, srednješolsko izobraževanje ter terciarno izobraževanje. V terciarno izobraževanje spada celotno visoko šolstvo.

Priložnostno učno okolje si lahko predstavljamo kot okolje, kjer učenje poteka v razmerah, ki niso primarno namenjene učenju in nimajo določenih ciljev. Pri tem ponavadi nihče ne poučuje in rezultat takega učenja ni javno veljavna kvalifikacija. Učenje poteka iz izkušenj s poslušanjem ali gledanjem. Učenje v prostem času, šport, delo itd. spadajo pod priložnostno učno okolje.

Neformalno (angl. Non-formal learning) učno okolje si lahko predstavljamo kot okolje med formalnim in priložnostnim učnim okoljem. Takšno okolje je prav tako namerno, planirano učenje in ima določene cilje, ki pa se praviloma ne konča s pridobitvijo veljavne kvalifikacije in spričevala. Pod neformalno učno okolje spadajo učni tečaji (računalniški, učenje jezikov ...), inštrukcije itd.

Učna okolja bi lahko dodatno razdelili na:

- tradicionalna učna okolja – recimo uporaba klasičnih učnih pripomočkov (knjige, učbeniki ...)
- elektronska multimedijška učna okolja – recimo uporaba sodobne računalniške tehnologije (računalnik, internet, mobilni telefon ...)

Elektronsko multimedijško učno okolje lahko umestimo v vse tri načine učenja. Lahko ga uporabimo kot pomoč med poukom v šoli, lahko ga uporabljamo v prostem času ali pa pri učenju jezikovnih tečajev. Takšno okolje prinaša nov pristop k učenju in bogatenju besedišča. Lahko se ponudi listanje po e-stranah učbenika in delovnega zvezka. Prikazuje se lahko grafično in zvočno podprte e-naloge, didaktične igre ...

#### 3.1 Zamisel izdelave

Pri načrtovanju novega multimedijskega učnega okolja se je treba najprej odločiti za ciljno skupino uporabnikov učnega okolja. Učno okolje, ki je namenjeno srednješolcem, ni primerno za učenje v vrtcu ali osnovni šoli. Pomembno je tudi vedeti, ali bo učno okolje namenjeno samostojni (učenci samostojno uporabljajo okolje) ali skupni uporabi, kjer je učitelj tisti, ki skrbi za pravilno uporabo.

Prednost takega e-gradiva je, da učitelj in učenec lahko pri pouku uporabljata gradivo na različne načine in ga prilagodita situaciji, potrebi in namenu učenja. Omogoča hitro in ilustrativno predstavitev gradiv, skrajša čas priprave in boljšo predstavljivost posameznih tem. Prinaša svež način učenja tujega jezika, boljšo motivacijo za sodelovanje in je prilagojen sodobnim didaktičnim pristopom.

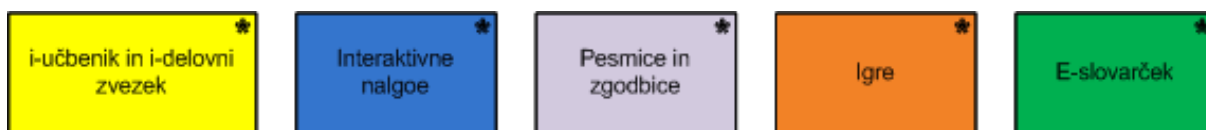
Pri načrtovanju učnega okolja smo za pomoč pri strukturiranju prosili učiteljico angleškega jezika v osnovni šoli, gospo Viljenko Šavli, ki je tudi soavtorica učbenika in delovnega zvezka MySails2 [25]. Struktura učnega okolja, ki ga bomo predstavili, je v celoti zgrajena na podlagi tega učnega gradiva in v sodelovanju z učiteljem. Na tak način smo s strokovno pomočjo sestavili okolje, ki je primerno zgrajeno za optimalno uporabo pri pouku.

Najprej se je potrebno odločiti, ali bomo v aplikaciji predstavili celotno učno snov učbenika ali samo del. V prikazanem zgledu, smo se odločili predstaviti strukturo aplikacije s samo enim poglavjem učbenika.

Aplikacijo smo strukturirali tako, da smo najprej s pomočjo učiteljice določili, kaj bo aplikacija v celoti omogočala, saj je potrebno istočasno predstaviti gradiva učencem in učitelju.

### 3.1.1 Razdelitev na več sklopov

Okolje je razdeljeno na več vsebinskih sklopov:



Slika 15: Deli okolja

Razdelitev gradiva v sklope je pogojena z vsebino in namenom didaktične uporabe gradiv pri pouku.

Prvi sklop »**i-učbenik in i-delovni zvezek**« omogoča učitelju jasen prikaz posamezne obravnavane strani v učbeniku in delovnem zvezku, možnost označevanja posameznih elementov za lažje razumevanj v procesu učenja, reševanje nalog in shranjevanje rešitev, učencem pa lažjo orientacijo, nazornejši prikaz in spremljanje gradiva ter možnost neposrednega in zanimivejšega reševanja interaktivnih nalog.

Drugi sklop »**Interaktivne naloge**« je didaktično namenjen individualnemu in skupinskemu interaktivnemu usvajanju, utrjevanju in ponavljanju snovi sklopa z neposredno računalniško povratno informacijo o pravilnosti odgovorov. Učitelju in učencu omogoča postopno in zahtevnosti ter spretnosti primerno uporabo. Poskrbljeno pa je tudi za samopreverjanje pridobljenega znanja.

Tretji sklop »**Pesmice in zgodbe**« je dodaten didaktičen pripomoček, ki ponuja učitelju možnost, da popestri in ponazori obravnavane vsebine z zvočno opremljeno animacijo. Poleg omenjenega ta sklop omogoča tudi različne didaktične pristope, tako lahko za ponavljanje

učencem prikažemo naključno izbrane dele animacije ali pa izključimo zvok in učenci z igro vlog ali petjem sodelujejo ob animaciji.

Četrty sklop »**Igre**« ponuja raznolike didaktične igre, ki so v bistvu namenjene utrjevanju in preverjanju snovi, so pa izdelane v obliki računalniških iger, in zato za učence veliko bolj zanimive in motivacijske. Z njimi poskušamo ustvariti sproščenost in narediti učenje zanimivejše in igrivo.

Peti sklop »**Slovar**« je na tej stopnji, pa tudi sicer pri učenju tujih jezikov, nepogrešljiv dodatek. Učencu daje možnost sprotnega preverjanja pomena novih besed v vseh sklopih in zato uspešnejšega učenja, saj slovar zajema celoten nabor besed iz učbenika, ki je abecedno urejen in vedno dostopen.

#### **a) i-učbenik in i-delovni zvezek**

Za interaktiven učbenik in delovni zvezek (angl. Interactive Book – i-Book) je potrebno pripraviti ustrezno gradivo. V kolikor se izdeluje gradivo na podlagi klasičnega gradiva, lahko ta gradiva skeniramo v elektronsko obliko (sliko) ali pa gradivo izdelamo sami. Takšno gradivo lahko s prej opisanimi orodji nato ustrezno obdelamo in prikažemo. Ker smo želeli zanimiv in uporaben prikaz strani učbenika, smo za prikaz uporabili orodja iSPRING. Vgrajen predvajalnik ima dodatne možnosti risanja, označevanja, postavitve v celozaslonski način, itd.

Pri e-učniku smo prišli tudi na idejo, da uporabimo dodatno možnost izdelave interaktivnih kvizov, ki jo ponuja orodje iSPRING. Ob prikazovanju posameznih strani učbenika smo dodali pripadajoče interaktivne naloge za utrjevanje snovi. Posamezno poglavje učbenika smo predstavili v sklopih po dve strani skupaj, vsak tak sklop pa ima dodan niz interaktivnih nalog, ki je povezan z vsebino na teh straneh.

Med posameznimi sklopi se bodo učenci prosto gibal, zato smo dodali možnost prehajanja »naprej« in »nazaj« ter možnost izbiranja med vsemi sklopi. Več o tem bomo opisali v poglavju 2.3.

#### **b) Interaktivne naloge**

Učenec se s pomočjo interaktivnih vsebin uči, utrjuje in preverja svoje znanje. Za vsako poglavje smo pripravili štiri tematske sklope interaktivnih nalog (angl. Interactive exercises), ki so povezane z vsebino poglavja v učbeniku, in en preizkus znanja. Interaktivne naloge so zgrajene tako, da prehajajo od lažjih k težjim. Omogočajo spremljanje uspešnosti reševanja, možnost neomejenega ponavljanja, so bogato opremljene z ilustracijami iz učbenika in zvočnimi posnetki ter omogočajo nadgrajevanje spretnosti branja, poslušanja in pisanja. Z njimi učenec utrjuje besedišče, tvori povedi, ureja pesmice in dopolnjuje krajše zgodbe. Interaktivne naloge omogočajo izbiro različnih tipov nalog. Uporabili smo naloge povezovanja, dopolnjevanja, spomina, magičnega kvadrata, uganke, dopolnjevanje praznin, urejanje besedila in križanke.

Preizkus znanja je namenjen preverjanju, koliko se je v tem poglavju učenec naučil. Naloge v testu so različnega tipa: branje, pisanje in poslušanje. Po sprožitvi testa učenec zapiše v obrazec svoje ime, ob zaključku pa dobi pregled odgovorov in oceno uspešnosti reševanja, ki ga lahko tudi natisne.

Za izdelavo interaktivnih nalog smo uporabili orodje jCLIC, pri izdelovanju preizkusa znanja pa smo si pomagali z orodjem Quizfaber.

### **c) Pesmice in zgodbe**

V učbeniku imajo pomembno mesto pesmice in zgodbe (angl. Songs And Stories), ki učencem omogočajo lažje usvajanje določenih vsebin. Pesmice z rimo pomagajo učencem usvojiti posamezne tematske vsebine, zgodbe pa predstavijo temo v daljšem besedilu in učence navajajo na poslušanje in sklepanje o vsebini iz dogajanja samega. Prav zaradi teh dejstev smo se odločili v posameznem poglavju dodati še animacije za dve pesmici in zgodbico. Tako učenci ob poslušanju in opazovanju dogajanja lažje spremljajo in razumejo vsebino. Pri tem smo uporabili obstoječe ilustracije pesmic in zgodbe in jih prikojili za potrebe animacij v programu PowerPoint.

Pri iskanju rešitve za izdelavo animiranih pesmic in zgodbic smo se spet poslužili orodja iSPRING, ki se vgradi v okolje PowerPoint. Orodje je žal plačljivo, vendar ga je mogoče preizkusiti tudi brezplačno.

S pomočjo orodja iSPRING je izdelovanje animiranih pesmic in zgodbic enostavno, saj podpira vse možnosti PowerPoint-a. Treba je le izdelati predstavitev z različnimi učinki, ki jo nato lahko izvozimo v tehnologijo Flash.

### **d) Igre**

Učenje z računalnikom na tej starostni stopnji je podobno igri, zato smo se odločili, da poleg učenja z interaktivnimi nalogami ponudimo še utrjevanje znanja s posebnimi didaktičnimi igrkami (angl. Games). Te tudi preverjajo znanje, vendar v preobleki znanih računalniških iger ali animacij. Motivacija pri reševanju je različna: zabijanje golov, zadeti koš ali preverjanje dobrega spomina in tekmovanje s časom. Izgled in grafična podoba je učencem znana in zato tudi bolj všečna.

Igre so izdelane s pomočjo orodij Spellmaster, Memory Maker in Content Creator. Nekaj iger smo izdelali tudi samostojno s pomočjo programskega jezika Action Script 3 ter Adobe Flash.

### **e) Slovar**

Ker je na tej stopnji učenje angleščine šele na začetku, učenec potrebuje tudi pomoč pri prevodu posameznih izrazov. Zato smo se odločili, da ponudimo angleško-slovenski elektronski slovar za besede in izraze, ki jih spoznajo v učbeniku.



S pomočjo slovarja (angl. Dictionary) si lahko učenec kadarkoli pomaga pri razumevanju raznih pojmov, ki se pojavljajo v besedilu. Učenec lahko vnose enostavno poišče po abecednem kazalu. Slovar je lahko ves čas dela z iBookom odprt in dostopen.

E-slovar je izdelan s pomočjo licenčnega orodja Articulate Engage, s katerim lahko izdelamo veliko različnih interaktivnih flash animacij, med njimi tudi slovar. Tudi to orodje je žal plačljivo, vendar ga je mogoče preizkusiti tudi brezplačno in z njegovo pomočjo izdelati posamezne krajše uporabne izdelke.

### **3.1.2 Struktura uporabniškega vmesnika**

Uporabniški vmesnik bo sestavljen iz 3 nivojev: izbira med različnimi sklopi (i-učbenik, i-naloge, igre ...), izbira učne vsebine in prikaz vsebine.

Učenec ima možnost izbiranja med 5 različnimi sklopi gradiv, ki smo jih opisali v poglavju 2.2. S takim načinom izbiranja ima učenec pregled nad posameznimi sklopi, ki jih enota ponuja.

Ob izbiri enega od sklopov, se učencu prikaže vsebina trenutno izbranega sklopa. Glede na to, da ima vsak različno število gradiv, je potrebno za vsak sklop izdelati svoj vmesnik.

Pri izbiri i-učbenika in i-delovnega zvezka, ima učenec na voljo izbiro med različnimi stranmi učbenika in izbiro delovnega zvezka. Pri učbeniku je na voljo 8 gumbov (vsak izmed gumbov predstavlja 2 strani učbenika), pri delovnem zvezku pa en gumb.

Pri interaktivnih nalogah se izbira med petimi gumbi, za pesmice in zgodbe so 3 gumbi, igre pa učenci izbirajo med 8 gumbi, ki jih bodo pripeljali do izbrane naloge, igre, zgodbe ali preizkusa znanja.

V primeru izbire slovarja se izvede samostojna aplikacija, ki prikaže slovar v svojem oknu.

Pri prikazu vseh učnih vsebin ima učenec na voljo izbiro naslednje in predhodne naloge, vpogled v slovar ter ponovno možnost izbire med vsemi nalogami trenutno izbranega nivoja.

## **3.2 Opis rešitve**

Predstavili smo tehnologije učnega okolja in multimedijška učna orodja za izdelavo multimedijških učnih vsebin. Opisali smo tudi metodologijo programiranja in s pomočjo strokovne pomoči predstavili zamisel izdelave multimedijškega učnega okolja.

Sedaj bomo predstavili praktični primer izdelave multimedijškega učnega okolja na podlagi vseh prej omenjenih tehnologij in orodij. Opisali bomo podatkovni model, sestavo aplikacije, uporabniške vmesnike in uporabljene podatkovne strukture.

### **3.2.1 Podatkovni model**

Konceptualni model podatkovne baze smo izdelali z orodjem SQL Power Architect [21]. Prikazuje le entitetne tipe, ki vsebujejo ime, enolični ključ in attribute.

Podatkovni model (slika 16) je sestavljen iz tabel `ibook`, `exercises`, `games` in `songsStories`. Te tabele vsebujejo vse podatke o učnih vsebinah v posameznem sklopu. V času izdelave učnega okolja je večkrat prišlo do želje po spremembi določenega sklopa, zato ima vsak sklop zaradi robustnosti pri dodajanju novih zahtev svojo tabelo. Tako lahko v aplikacijo hitreje dodajamo in odstranjujemo funkcionalnosti posameznih sklopov. V primeru uporabe skupne tabele bi bilo potrebno za vsako majhno spremembo preveriti vse sklope in testirati združljivost.

<b>ibook</b>	<b>songsStories</b>
id: INTEGER [PK]	id: INTEGER [PK]
name: VARCHAR(50)	name: VARCHAR(50)
opis: VARCHAR(250)	opis: VARCHAR(250)
kvizUrl: VARCHAR(250)	ssUrl: VARCHAR(250)
ucbenikUrl: VARCHAR(250)	
ucbenikThumb: VARCHAR(250)	
<b>exercises</b>	<b>games</b>
id: INTEGER [PK]	id: INTEGER [PK]
name: VARCHAR(50)	name: VARCHAR(50)
opis: VARCHAR(250)	opis: VARCHAR(250)
exercisesUrl: VARCHAR(250)	gameUrl: VARCHAR(250)
	button_number: INTEGER

Slika 16: Podatkovni model

## 1. iBook

Enolični ključ entitetnega tipa `iBook` je `id`, ki praktično pomeni zaporedno številko prikazanega sklopa strani iz učbenika.

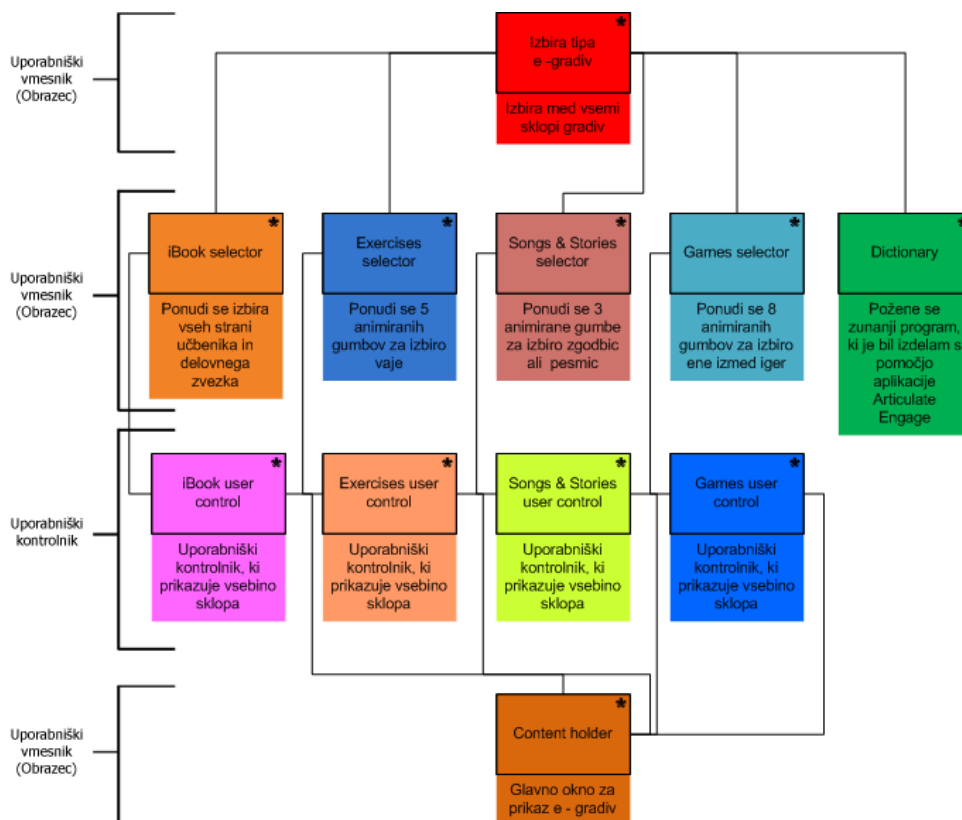
Atribut `name` predstavlja ime trenutnega sklopa (v našem primeru je to `Stran 1-2`, `Stran 3-4` itd.). Opis predstavlja kratek opis trenutnega sklopa, medtem ko `kvizUrl` predstavlja fizično pot do elektronskega gradiva na disku, ki se prikaže ob izbiri. `ucbenikThumb` predstavlja fizično pot do predogledne slikice strani učbenika, ki si jo uporabnik ob izbiri lahko ogleda v povečani različici. `ucbenikUrl` pa predstavlja fizično pot do elektronskega gradiva, ki predstavlja skenirani strani izbranega sklopa.

## 2. songsStories, exercises in games

V vseh treh entitetah sta skupna atributa `name`, `opis`, ki predstavljata ime, opis. Fizične poti do elektronskih gradiv so označene z atributi `ssUrl`, `exercisesUrl` in `gameUrl`. Posebnost entitete `Games` je atribut `button_number`, ki predstavlja gumb, preko katerega bo izbrano gradivo dosegljivo.

### 3.2.2 Izdelava aplikacije

V poglavju 3.1.1 smo multimedijško učno okolje razdelili na 5 vsebinskih sklopov. Za vsak sklop ima aplikacija izdelan svoj uporabniški vmesnik oziroma obrazec, preko katerega lahko učenec ali učitelj izbira med izdelanimi multimedijškimi učnimi vsebinami (slika 17). Učno okolje ima ob obrazcih za izbiro učnih vsebin posameznega sklopa še 2 dodatna obrazca. Prvi obrazec učencu ali učitelju ponudi izbiro (v obliki gumbov) med posameznimi sklopi, drugi pa prikazuje izbrano učno vsebino.



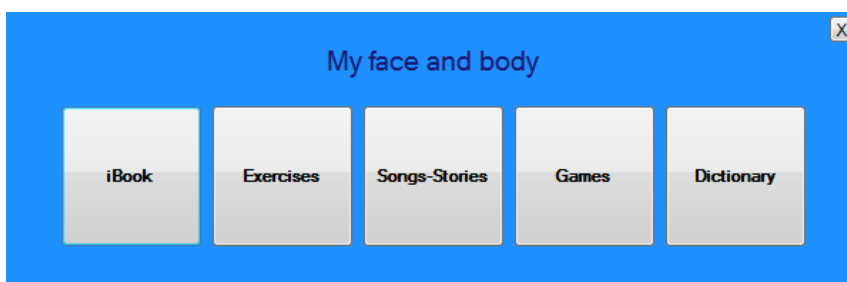
Slika 17: Sestava aplikacije

Aplikacija deluje tako, da se ob zagonu prikaže glavno izbirno okno. Učenec ali učitelj ima na voljo izbiro posameznega vsebinskega sklopa. Ob izbiri sklopa se prikaže obrazec, ki ponudi izbiro med vsemi učnimi vsebinami izbranega sklopa. Ko se učno vsebino izbere, se prikaže obrazec, ki to vsebino prikaže.

Obrazec, ki prikazuje učno vsebino, deluje v povezavi z uporabniškimi kontrolniki (angl. User control). Z uporabo uporabniških kontrolnikov za prikaz učnih vsebin, dobimo obrazec, ki ga lahko enostavno uporabimo za vse sklope. Lahko bi si predstavljali, da je podoben princip pri televizorju, kjer je okvir televizorja vedno isti, le slika se spreminja glede na izbrani program.

#### 3.2.2.1 Uporabniški vmesniki

Prvi obrazec preko gumbov ponudi izbiro med vsemi vsebinskimi sklopi (slika 18). Gumbi so dodani statično in se ne dodajajo avtomatično glede na vnose v podatkovni bazi. Tak način smo izbrali, ker se število izbir ne spreminja pogosto. V kolikor pride do spremembe je smotrnejše dodati novo izbiro ročno, kot pa izdelati sistem, ki bo avtomatično dodajal gumbke.



Slika 18: začetno okno

Glavne značilnosti prvega obrazca so izbira in prikaz izbranega vsebinskega sklopa. Obrazec je izdelan v programskem razredu `mainForm`, ki deduje iz razreda `Form`. Ta ima nadzor nad prikazom oken.

V inicializaciji razreda `mainForm` se deklarirajo vse potrebne spremenljivke, uvozijo pa se tudi Windows funkcije, ki skrbijo za premikanje obrazca z miško (koda 1) [13].

```
public const int WM_NCLBUTTONDOWN = 0xA1;
public const int HT_CAPTION = 0x2;

[DllImportAttribute("user32.dll")]
public static extern int SendMessage(IntPtr hWnd, int Msg, int wParam, int lParam);
[DllImportAttribute("user32.dll")]
public static extern bool ReleaseCapture();
```

Koda 1: Uvoz Windows funkcij

Ob klicu konstruktorja `mainForm()` se inicializirajo vse potrebne komponente in kreirajo vsi objekti, ki jih bomo potrebovali ob klicu oz. prikazu posameznega vsebinskega sklopa (koda 2). Ob tem se nastavi tudi začetno pozicijo obrazca `StartPosition` ter doda rutina za obravnavo dogodkov (angl. Event handler). Ta recimo obravnava pritisk na levo miškino tipko, s katerim lahko nato premikamo okno obrazca ob kliku na obrazec.

```
public MainForm()
{
    InitializeComponent();
    this.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;
    iBook_atributi = new iBook_attributes();
    iBook_atributi.setFrmMain = this;

    Games_atributi = new Games_attributes();
    Games_atributi.setFrmMain = this;

    ss_atributi = new SS_attributes();
    ss_atributi.setFrmMain = this;

    exercises_atributi = new Exercises_attributes();
    exercises_atributi.setFrmMain = this;

    this.MouseDown += new MouseEventHandler(MainForm_MouseDown);
}
```

Koda 2: Klic konstruktorja `mainForm()`

Vsak izmed razpoložljivih gumbov ima svojo rutino za dogodke (angl. Event handler), ki se sproži ob kliku nanj (koda 3). Ob kliku na posamezen gumb (vsi gumbi delujejo po istem principu, le drugi obrazci se inicializirajo) se sprva preveri, če izbrani vsebinski sklop (obrazec) že obstaja (to bi se zgodilo, če bi že imeli prikazan izbrani obrazec in bi ga želeli še

enkrat). Če je že prikazan, se ga s klicem na funkcijo `Close()` najprej zapre in nato kreira novega. V primeru da še ni prikazan, pa le kreiramo izbrani obrazec in ga prikažemo s pomočjo funkcije `Show()`. Ob kreiranju objekta `gameSelect` se kot parameter poda tudi objekt `Games_atributi`, ki je tipa `Games_atributi`, ki je pravzaprav podatkovna struktura za branje in upravljanje z izbranimi tipi nalog.

```
private void btnGames_Click(object sender, EventArgs e)
{
    if (gameSelect != null)
    {
        gameSelect.Close();
        gameSelect = new game_select(Games_atributi);
        gameSelect.Show();
    }
    else
    {
        gameSelect = new game_select(Games_atributi);
        gameSelect.Show();
    }
    this.Hide();
}
```

**Koda 3: Rutina dogodka za prikaz željenega vsebinskega sklopa**

Parameter se pošilja, ker je znotraj parametra sklic na glavno okno `mainForm`. To potrebujemo ob zapiranju trenutno izbranega sklopa (v tem primeru sklopa `Games`).

Posebnost pri tem obrazcu je zagon programa, ki prikazuje slovar (angl. Dictionary). Tega se zažene s pomočjo razreda `Process`, s katerim lahko poženemo katerokoli aplikacijo, ki jo imamo na razpolago v operacijskem sistemu (koda 4).

```
Process pr = new Process();

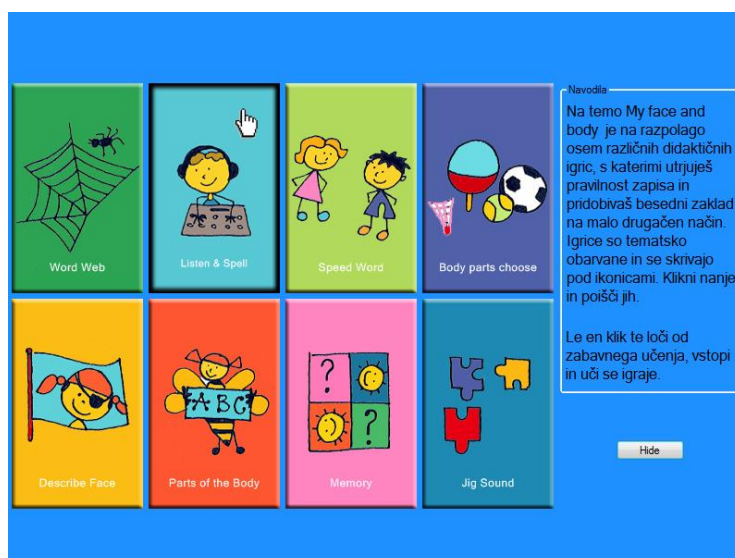
pr.StartInfo.FileName = @Application.StartupPath + @"\" + "1\\Dictionary\\Launch_Interaction.exe";

pr.Start();
```

**Koda 4: Primer zagona zunanjega procesa**

Z uporabo znaka `»@«` pred nizom (angl. Verbatim string literal) se znebimo uporabe znakov `»\, "«` itd. Znak `»\«` bi sicer morali uporabiti kot `»\\«`.

Ob kliku na gumb `Games` na prejšnem obrazcu, se nam prikaže nov obrazec, ki ponudi izbiro med različnimi multimedijskimi učnimi vsebinami (slika 19).



Slika 19: Okno za izbiranje željene igrice

Programsko okno, ki se prikaže, je sestavljeno iz razreda `game_select` in razreda `mainForm`. Uvožita se Windows funkciji `SendMessage()` in `ReleaseCapture()`, ki skrbita za premikanje obrazca z miško.

Konstruktor, ki se kliče ob inicializaciji, nastavi vse potrebne komponente (funkcija `InitializeComponent()`), shrani podani parameter `Games_attributes`, kreira rutino dogodka pritiska na miškino tipko (angl. `MouseEventHandler`) ter inicializira objekt `graphicUtilities`, ki je tipa `graphicUtilities()` in se uporablja za grafično obdelavo slik.

```
public game_select(Games_attributes Games_attributes)
{
    this.DoubleBuffered = true;
    InitializeComponent();
    this.Games_attributes = Games_attributes;
    this.MouseDown += new MouseEventHandler(game_select_MouseDown);
    graphicUtilities = new graphicUtilities();
}
```

Koda 5: Klic konstruktorja

Ob naložitvi (angl. `Load`) okna se kliče funkcija `game_select_Load()`, ki nastavi začetno pozicijo okna z `setFormPosition()` ter animacijo okna ob prikazu, inicializira objekte za delo z bazo z `loadFromDataBase()` ter naloži vse slike, ki so uporabljene kot gumbi za prikaz izbrane naloge s `setPictures()` (koda 6).

```
private void game_select_Load(object sender, EventArgs e)
{
    setFormPosition();
    loadFromDataBase();
    setPictures();
}
```

Koda 6: Ob dogodku naložitve (angl. `Load`)

Posebnost funkcije `setFormPosition()` je animiranje začetnega prikaza okna. Za animiran prikaz okna izkorišča Windows funkcijo `AnimateWindow()` [6].

Funkcija ima 3 vhodne parametre `IntPtr hwnd`, `int dwTime`, `int dwFlags`. Prvi parameter je tipa `IntPtr`, ki predstavlja kodo za dostop (angl. Handle) do upravljalca komponente v našem primeru upravljelec okna `this.Handle`. Parameter `dwTime` predstavlja čas animacije v milisekundah, `dwFlags` pa predstavlja zastavice (angl. Flags), ki povedo, kaj naj se v animaciji zgodi.

Primer uporabe animacije, ki prikaže okno z animacijo diapozitivov. Uporabi se efekt pojemanja od zgoraj proti dnu (koda 7).

```
WinApi.AnimateWindow(this.Handle, 500, WinApi.AW_ACTIVATE | WinApi.AW_BLEND | WinApi.AW_SLIDE | WinApi.AW_VER_POSITIVE);
```

**Koda 7: Primer uporabe funkcije za animacijo**

S pomočjo te funkcije lahko animirano okno prikažemo ali skrijemo. Možnosti so naslednje:

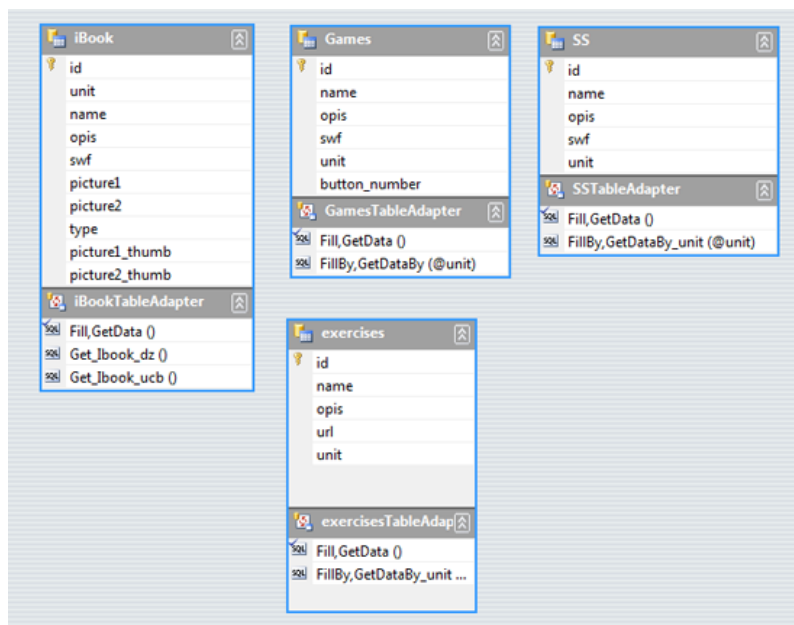
- Animacija okna horizontalno od leve proti desni.  
`int AW_HOR_POSITIVE = 0x1;`
- Animacija okna horizontalno od desne proti levi.  
`int AW_HOR_NEGATIVE = 0x2;`
- Animacija okna vertikalno od zgoraj proti dnu.  
`int AW_VER_POSITIVE = 0x4;`
- Animacija okna vertikalno od spodaj proti gor  
`int AW_VER_NEGATIVE = 0x8;`
- V primeru uporabe zastavice `AW_HIDE` se okno zapre samo vase, če `AW_HIDE` zastavica ni uporabljena, pa se okno prikaže iz centra v vse smeri.  
`int AW_CENTER = 0x10;`
- Funkcija skrije prikazano okno  
`int AW_HIDE = 0x10000;`
- Funkcija prikaže okno  
`int AW_ACTIVATE = 0x20000;`
- Animacija se prikaže kot animacija diapozitivov (angl. Slide animation)  
`int AW_SLIDE = 0x40000;`
- Animacija se prikaže z uporabo efekta pojemanja (angl. Fade effect)  
`int AW_BLEND = 0x80000;`

V metodi `loadFromDataBase()` se inicializira objekt `GamesAdapter`, ki je tipa `GamesTableAdapter` (koda 8). Ta tip objekta je kreiran na podlagi `DataSet` komponente .NET, ki skrbi za povezavo med bazo in aplikacijo.

```
private void loadFromDataBase()
{
    GamesAdapter = new MySails.EntriesDataSetTableAdapters.GamesTableAdapter();
    _Games_attributes.setData = GamesAdapter.GetGameData(1);
}
```

**Koda 8: Klic funkcije `loadFromDataBase()`**

Znotraj komponente `DataSet` deklariramo tabele (slika 20), ki so enake tabelam v podatkovni bazi. Na vsako od deklariranih tabel lahko dodamo možnosti dostopa, ki s pomočjo ustreznih sql stavkov priskrbijo podatke iz podatkovne baze.



Slika 20: Kreiran podatkovni tip (angl. DataSet)

Ko se adapter inicializira, se z adapterjem `GetGameData()` preko objekta `_Games_attributes` napolni podatkovna tabela. Adapter kot rezultat funkcije vrne podatkovno tabelo, v kateri so vsi potrebni podatki, ki jih aplikacija potrebuje.

Metoda `setPictures()` skrbi za nastavitve slik, ki se uporabljajo kot gumb.

```
pb1.Image = graphicUtilities.setTextOnPicture(_Games_attributes.getName(1), "MySails.picturesGames.1.png", false, false);
pb1.MouseEnter += new EventHandler(mouse_enter);
pb1.MouseLeave += new EventHandler(mouse_leave);
pb1.Click += new EventHandler(mouse_click);
```

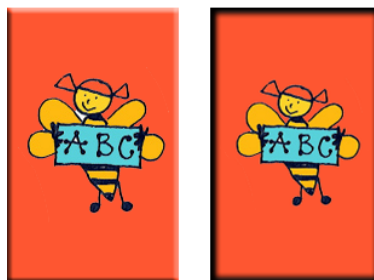
#### Koda 9: Primer nastavitve

Primer v kodi 9 prikazuje nastavitve slikovnega okna (angl. Picture box) `pb1`. Za prikaz željene slike je potrebno nastaviti spremenljivko `Image`, ki sprejme sliko tipa `Bitmap`, ki smo jo kreirali s pomočjo objekta za grafično obdelavo (`graphicUtilities`) s klicem funkcije `setTextOnPicture`.

Ker se slikovno okno uporablja kot animiran gumb, je potrebno dodati več rutin dogodkov (angl. Event Handler), ki bodo skrbeli za vse potrebne dogodke, da bo slika izgledala kot gumb.



Za avtentični prikaz gumba smo izdelali dve različni sliki, normalno in sliko izbranega gumba (slika 21).



Slika 21: Različni sliki za izdelavo gumba

Dogodka `MouseEnter` in `MouseLeave` skrbita za pravilen prikaz izdelanih slik. Prvi dogodek se sproži, ko miška vstopi v slikovno okno, drugi pa takrat ko miška zapusti slikovno okno. Oba dogodka kličeta funkcijo `leaveEnter`, vendar z različnimi vhodnimi parametri `EventType`, ki pove, če je miška nad sliko in če je sliko zapustila.

```
private void leaveEnter(String sender,int EventType)
{
    if (EventType == 1)
    {
        pbl.Cursor = Cursors.Default;
        pbl.Image = graphicUtilities.setTextOnPicture(_Games_attributes.getName(1), "MySails.picturesGames.1.png", false, false);
    }
    else if (EventType == 2)
    {
        pbl.Cursor = Cursors.Hand;
        pbl.Image = graphicUtilities.setTextOnPicture(_Games_attributes.getName(1), "MySails.picturesGames.1_down.png", true, false);
    }
}
```

Koda 10: Funkcija `leaveEnter`

V funkciji (koda 10) se nastavi miškin kazalec glede na to, ali je miška nad sliko ali ne, ter pravilno prikaže sliko. V kolikor se sproži dogodek `MouseEnter`, se prikaže slika, ki prikazuje vdrtlo sliko. To se zgodi s klicem funkcije `setTextOnPicture`, ki nastavi napis na sliko ter prikaže pravilno sliko glede na vhodne parametre.

Funkcija `setTextOnPicture` sprejme 4 vhodne parametre:

- `textToWrite` (tipa `string`) - napis, ki ga želimo dodati na sliko
- `path` (tipa `string`) - pot do slike, ki jo želimo kopirati in popraviti
- `small` (tipa `boolean`) - pove ali se napis izrisuje na sliko, ki prikazuje gumb v stanju, ko je miška postavljena nad njim ali ko ga je zapustila
- `landscape` (tipa `boolean`)- pove ali je slika ležeča ali pokončna, ker so nastavljene različne velikosti pisave

Za vsak gumb se definira tudi dogodek, ko z miško kliknemo nanj. Ta dogodek je potrebno ujeti in poskrbeti za prikaz izbrane igrice. Ko se dogodek `Click` sproži, kliče funkcijo, ki prikaže izbrano igrico na novem obrazcu.

```

void mouse_click(object sender, EventArgs e)
{
    PictureBox pictureClicked = (PictureBox)sender;
    int pictureID = int.Parse(pictureClicked.Name.Substring(2, 1));

    _Games_attributes.setCurrentGame((pictureID-1));

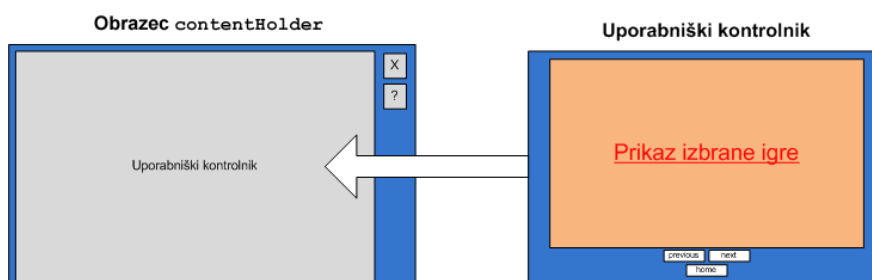
    this.Hide();
    contentHolder content = new contentHolder(_Games_attributes);
    content.getParrent = this;
    content.Show();
}

```

**Koda 11: Dogodek klika na miškino tipko**

Posebnost te funkcije (koda 11) je prikaz igre, ki jo je uporabnik izbral ob kliku na določen gumb. Prikaz se vrši tako, da na obrazec `contentHolder` preko podatkovne strukture `_Games_attributes` pošljemo podatke o vseh igrah (recimo podatki o tem, kje na disku se igra nahaja), ki smo jih dobili ob poizvedbi na podatkovno bazo. Vse podatke pošiljamo zato, ker se v funkciji `setCurrentGame`, ki smo jo klicali pred tem, nastavi vse potrebne spremenljivke za izbrano igro in preveri, če v zaporednem vrstnem redu obstaja še kakšna igra za in pred trenutno. To pa je potrebno zato, ker na obrazcu obstaja povezava na predhodno in naslednjo igro, če ta obstaja.

Obrazec `contentHolder` je programsko okno, ki je sestavljeno tako, da ga lahko uporabljamo za vse sklope učnih vsebin, ki jih uporabljamo v aplikaciji: igre, pesmice, zgodbe (slika 22). Vse kar je potrebno, je narediti uporabniški kontrolnik (angl. User control) za vsak sklop učnih vsebin.



**Slika 22: Prikaz delovanja obrazca**

Delovanje obrazca je podobno kot za ostala okna, ki smo jih opisali. Uvozi `Windows` funkcijo za premikanje okna ob kliku z miško, kreira animacijo s pomočjo funkcije `AnimateWindow` ter nastavi začetno pozicijo okna.

Pri opisovanju programskega okna `games_select` je bilo prikazano, kako se `contentHolder` prikaže (koda 12).

```

contentHolder content = new contentHolder(_Games_attributes);
content.Show();

```

**Koda 12: Prikaz okna contentHolder**

Razred `contentHolder` vsebuje toliko konstruktorjev kolikor je različnih vsebinskih sklopov. Razlikuje se v tipu vhodnega parametra: če želimo prikazati učno vsebino `Games` pošljemo kot vhodni parameter podatkovno strukturo, ki jo ta uporablja (koda 13).

```
//klic iz GAMES
public contentHolder(Games_atributes Games_atributes)
{
    InitializeComponent();
    this._Games_atributes = Games_atributes;
    this._MouseDown += new MouseEventHandler(contentHolder_MouseDown);
    pnlContent.MouseDown += new MouseEventHandler(contentHolder_MouseDown);

    setFormPosition();
    showGames();
}
```

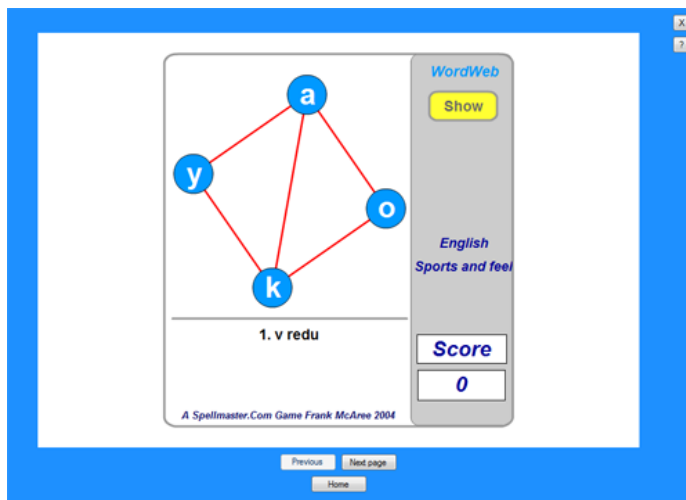
Koda 13: Primer konstruktorja za učno vsebino Games

V konstruktorju se tako kot drugje inicializirajo vse potrebne komponente, shrani se vhodni parameter, ki predstavlja podatkovno strukturo izbrane učne vsebine, kreira se rutina dogodka za klik na miškino tipko, animira prikaz okna, nastavi začetno pozicijo ter na okno prikaže (slika 23) izbrano učno vsebino (klic metode `showGames()`).

```
private void showGames()
{
    Games GamesControl = new Games(_Games_atributes);
    this.pnlContent.Controls.Clear();
    this.pnlContent.Controls.Add(GamesControl);
    GamesControl.Dock = DockStyle.Fill;
}
```

Koda 14: Klic metode `showGames()`

Metoda (koda 14) inicializira uporabniški kontrolnik `Games`, ki prikazuje učno vsebino. Inicializiran uporabniški kontrolnik se nato doda na delovno ploščo (angl. Panel) in razporedi čez celotno ploščo z ukazom `GamesControl.Dock = DockStyle.Fill;`



Slika 23: Primer obrazca s prikazano igro

Uporabniški kontrolnik `Games` za prikaz učne vsebine uporablja dva načina. Prvi način je prikaz igre v tehnologiji Flash s pomočjo komponentnega objektnega modela (angl. Component Object Model - COM), drugi pa je prikaz preko komponente integrirane `WebBrowser`, ki predstavlja internetni brskalnik, s katerim lahko prikazujemo spletne aplikacije.

Ena ali druga rešitev se uporabi glede na ime datoteke s končnico, ki jo dobimo preko spremenljivke v podatkovni strukturi `_Games_attributes.setUrl`. Končnica datoteke `swf` zahteva komponento Flash, končnica `html` pa `WebBrowser`.

Primer uporabe komponente Flash (koda 15):

```
axShockwaveFlash1.BackgroundColor = 2003199;
axShockwaveFlash1.AllowFullScreen = "true";
axShockwaveFlash1.SetVariable("quality", "high");
axShockwaveFlash1.Name = "test";
axShockwaveFlash1.Movie = @Application.StartupPath + "" + _Games_attributes.setUrl;

axShockwaveFlash1.Refresh();
WinApi.AnimateWindow(this.Handle, 100, WinApi.AW_HIDE | WinApi.AW_BLEND | WinApi.AW_VER_POSITIVE);
WinApi.AnimateWindow(this.Handle, 500, WinApi.AW_ACTIVATE | WinApi.AW_SLIDE | WinApi.AW_HOR_NEGATIVE);
```

**Koda 15: Primer uporabe komponente Flash**

Komponenti `axShockwaveFlash1` lahko nastavljamo tudi določene lastnosti:

- barva ozadja (RGB pretvorjena v desetiško število) `BackgroundColor`,
- omogočen celozaslonski prikaz `AllowFullScreen`,
- izbro kvalitete predvajanja,
- ime `Name` in
- predvajani film `Movie`

Primer uporabe komponente `WebBrowser` (koda 16):

```
okno.Hide();
String url = @Application.StartupPath + "" + _Games_attributes.setUrl;
url = url.Replace("\\", "/");
okno.Navigate(url);

WinApi.AnimateWindow(this.Handle, 100, WinApi.AW_HIDE | WinApi.AW_BLEND | WinApi.AW_VER_POSITIVE);
WinApi.AnimateWindow(this.Handle, 500, WinApi.AW_ACTIVATE | WinApi.AW_SLIDE | WinApi.AW_HOR_NEGATIVE);
okno.Show();
```

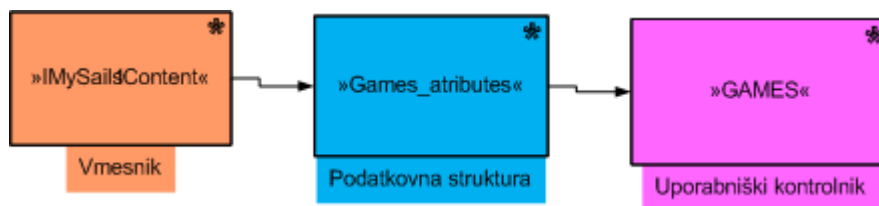
**Koda 16: Primer uporabe komponente WebBrowser**

Uporaba komponente `WebBrowser` je zelo enostavna. Za prikaz spletne strani ali spletnega mesta lahko enostavno vpišemo spletni naslov z navezavo na funkcijo `Navigate()`, ki kot vhodni parameter sprejme spletni naslov. Komponenta ima veliko možnih nastavitev, ki omogočajo programerju veliko možnosti za njegovo uporabo. Nekaj zanimivih in uporabnih nastavitev:

- `AllowNavigation` – Ali naj komponenta dovoli navigiranje iz trenutne strani na drugo
- `IsWebBrowserContextMenuEnabled` – Ali naj komponenta prikazuje kontekstni menu (desni miškin gumb) brskalnika
- `ScrollBarsEnabled` – Ali naj ima brskalnik omogočene drsne trakove (angl. Scroll bars)
- `WebBrowserShortcutsEnabled` – Ali naj komponenta omogoča uporabo bližnjic s pomočjo tipkovnice (npr: novo okno - CTRL+N)

### 3.2.3 Podatkovna struktura

#### Podatkovna struktura Games\_attributes



Slika 24: Diagram povezave uporabiškega kontrolnika z podatkovno strukturo

Podatkovna struktura Games\_attributes, ki jo uporablja uporabniški kontrolnik Games, je razred, ki vsebuje več spremenljivk ter funkcij, ki skrbijo za podatke o igrah.

Vse podatkovne strukture projekta uporabljajo vmesnik IMySailsContent (koda 17). Vmesnik izgleda kot navaden razred, vendar nima izvedbe. Vsebujejo le definicije dogodkov, metod ali lastnosti. Tako morajo vsi razredi, ki implementirajo vmesnik, obvezno uporabljati definicije vmesnika. S tem dobimo arhitekturo priključi-in-predvajaj (angl. Plug-and-play).

```

interface IMySailsContent
{
    void setNextPage();
    void setPreviousPage();
    String getName(int entry_id);
    void setCurrentGame(int entry_id);

    MainForm setFrmMain
    {
        get;
        set;
    }

    ostale definicije
}
  
```

Koda 17: Primer vmesnika

Pomembna metoda strukture je setCurrentGame(), ki glede na podan vhodni parameter nastavi vse potrebne podatke za izbrano igro.

```

public void setCurrentGame(int entry_id)
{
    setId = entry_id;
    setName = data[setId].name;
    setOpis = data[setId].opis;
    setUrl = data[setId].swf;

    setName = data[setId].name;
    //setNextPage
    //če obstaja naslednji vnos nastavimo _nextPage = true
    if ((setId + 1) < data.Count)
    {
        nextPage = true;
    }
    else
    {
        nextPage = false;
    }
    //setPreviousPage
    if ((setId - 1) >= 0)
    {
        previousPage = true;
    }
    else
    {
        previousPage = false;
    }
}

```

**Koda 18: Primer metode setCurrentGame()**

Metoda (koda 18) poleg podatkov o lokaciji, imenu, opisu in ostalih podatkov preveri, če obstaja še kakšna igra za ali pred trenutno izbrano igro. Tako lahko uporabnik enostavno prehaja iz ene igre v drugo, brez potrebe po zapiranju okna in ponovnem izbiranju igre preko glavnega obrazca. Preverjanje poteka tako, da kot vhodni parameter dobimo identifikacijsko številko izbrane igre. Preko te lahko ugotovimo, ali obstaja še kakšna igra za ali pred njo. Vseh iger je 8, zato številka pomeni, da naslednje igre ni, igra pred njo pa je igra pod številko 7. V kolikor obstaja, se nastavi zastavica `previousPage` (predhodnja igra) in `nextPage` (naslednja igra) in tako uporabniški kontrolnik `Games` omogoči ali onemogoči gumba `Previous` in `Next`.

## Razred za grafično obdelavo gumbov `graphicUtilities`

Funkcija `setTextOnPicture` se uporablja pri izrisovanju gumbov za izbiro iger. Razred `games_select` jo kliče iz različnih rutin dogodkov, ki so povezani s prikazom gumba (koda 19).

Delo funkcije je dodajanje napisa na sliko (slika 25), ki jo je funkcija dobila kot informacijo o lokaciji. Funkcija tudi razlikuje med ležečimi in pokončnimi slikami, prepozna pa tudi primer slike, ki prikazuje udrto sliko (pritisnjen gumb) ali normalno sliko. Razlike se nahajajo samo v velikosti črk, ki se dodajajo na sliko.

```

public Bitmap setTextOnPicture(string textToWrite, string path, bool small, bool landscape)
{
    Bitmap pic = new Bitmap(this.GetType().Assembly.GetManifestResourceStream(path));
    Graphics g = Graphics.FromImage(pic);
    StringFormat strFormat = new StringFormat();

    g.TextRenderingHint = System.Drawing.Text.TextRenderingHint.AntiAlias;
    if (landscape)
    {
        strFormat.Alignment = StringAlignment.Center;
        strFormat.LineAlignment = StringAlignment.Center;
        if (small)
            g.DrawString(textToWrite, new Font("Microsoft Sans Serif", 17, FontStyle.Bold), Brushes.White, new
RectangleF(0, 0, 227, 142), strFormat);
        else
            g.DrawString(textToWrite, new Font("Microsoft Sans Serif", 18, FontStyle.Bold), Brushes.White, new
RectangleF(0, 0, 227, 142), strFormat);
    }
    else
    {
        strFormat.Alignment = StringAlignment.Center;
        strFormat.LineAlignment = StringAlignment.Far;
        if (small)
            g.DrawString(textToWrite, new Font("Microsoft Sans Serif", 11, FontStyle.Bold), Brushes.White, new
RectangleF(0, -25, 142, 227), strFormat);
        else
            g.DrawString(textToWrite, new Font("Microsoft Sans Serif", 12, FontStyle.Bold), Brushes.White, new
RectangleF(0, -20, 142, 227), strFormat);
    }
    return pic;
}

```

Koda 19: Primer dodajanja teksta na sliko

Dodajanje napisa na sliko se vrši preko grafičnega razreda `Graphics`. Razred vsebuje veliko uporabnih metod in funkcij za delo s slikami. Za dodajanje napisa se uporablja `DrawString`.

```
Graphics.DrawString(s String, font Font, brush Brush, layoutRectangle RectangleF, format StringFormat)
```

Koda 20: Definicija metode `DrawString()`

Vhodni parametri so:

- `s`: napis, ki se doda na sliko,
- `font`: tip pisave, ki se uporabi za napis,
- `brush`: določi barvo pisave,
- `layoutRectangle`: pravokotnik na sliki, kjer se napis doda in
- `format`: objekt tipa `StringFormat` določi specifične attribute, ki urejajo postavitev napisa.

Funkcija kot rezultat vrne sliko v obliki bitne slike (angl. `Bitmap`), ki vsebuje dodan napis.



Slika 25: primer dodanega napisa na sliko

### 3.2.4 Praktični zgled

Primer uporabe aplikacije (slika 26):

- v prvem obrazcu izberemo gumb **Games**,
- v drugem obrazcu izberemo gumb **Parts of the body in**
- v zadnjem obrazcu se multimedijška učna vsebina prikaže.



Slika 26: Praktični zgled



## 4. ZAKLJUČKI

Za diplomsko nalogo smo si zastavili razvoj multimedijskega učnega okolja uporabnega pri pouku in doma, ki se je izkazala za smiselno in koristno. Ugotovili smo, da se lahko s pomočjo prostodostopnih multimedijskih orodij, enostavno izdelata zanimiva didaktična elektronska gradiva uporabna pri pouku različnih predmetov, ki pa jih je potrebno primerno urediti v smiselne sklope in ponuditi v ustreznem multimedijemskem učnem okolju. Tega pa trenutno še ni bilo na razpolago.

Multimedijско učno okolje se je že izkazalo kot uporabna oblika okolja za e-gradiva pri pouku angleščine, saj smo ga s pomočjo učiteljice Viljenke Šavli že tudi preizkušali v praksi. Predstavili smo ga nekaterim učiteljem angleščine na osnovnih šolah, ki so bili nad izvedbo, uporabnostjo in vsebino učnega okolja navdušeni in zadovoljni, saj trenutno nimajo na razpolago podobnega učnega okolja, ki bi ga lahko uporabili na i-tabli ali na računalnikih med poukom angleščine.

V bodoče bi lahko opisano multimedijско učno okolje še nadgradili, oziroma izboljšali. V opisanem primeru je multimedijška učna vsebina statično povezana v učno okolje, možna nadgradnja bi omogočala izdelati takšen vmesnik, ki bi posamezniku omogočal v učno okolje samostojno dodajati in odstranjevati učne vsebine. Ta možnost, bi spodbujala učitelja, da izdelata lastna elektronska učna gradiva, jih enostavno vključi v okolje in jih sproti dopolnjuje in nadgrajuje. Učiteljem bi ponudili izbiro že vnaprej izdelanih grafičnih predlog, s katerimi bi lahko prehajali med posameznimi vsebinskimi sklopi in učnimi vsebinami.

## 5. VIRI

- [1] (2010) Adobe Flash. Dostopno na:  
[http://en.wikipedia.org/wiki/Adobe\\_Flash](http://en.wikipedia.org/wiki/Adobe_Flash)
- [2] (2010) Advance Distributed Learning. Dostopno na:  
<http://www.adlnet.gov>
- [3] (2010) Agile software development. Dostopno na:  
[http://en.wikipedia.org/wiki/Agile\\_software\\_development](http://en.wikipedia.org/wiki/Agile_software_development)
- [4] (2010) Agilne metode razvoja programske opreme. Dostopno na:  
[http://sl.wikipedia.org/wiki/Agilne\\_metode\\_razvoja\\_programske\\_opreme](http://sl.wikipedia.org/wiki/Agilne_metode_razvoja_programske_opreme)
- [5] (2010) C Sharp (programming language). Dostopno na:  
[http://en.wikipedia.org/wiki/C\\_Sharp\\_%28programming\\_language%29](http://en.wikipedia.org/wiki/C_Sharp_%28programming_language%29)
- [6] (2009) C# Form Animation with Windows API. Dostopno na:  
<http://www.codeproject.com/KB/cs/FormAnimation.aspx?msg=3374144>
- [7] (2010) C# Overview. Dostopno na:  
[http://en.csharp-online.net/CSharp\\_Overview](http://en.csharp-online.net/CSharp_Overview)
- [8] Dr. Ivan Gerlič, Nekatera didaktična vprašanja vnašanja multimedije v izobraževanje. Maribor, Raziskovalni inštitut Pedagoške fakultete v Mariboru, 1997, str. 4.
- [9] (2010) E-gradiva izdelana preko razpisov ministrstva za šolstvo in šport. Dostopno na:  
[http://www.mss.gov.si/si/delovna\\_podrocja/ikt\\_v\\_solstvu/e\\_gradiva/](http://www.mss.gov.si/si/delovna_podrocja/ikt_v_solstvu/e_gradiva/)
- [10] (2010) Flash Player Version Penetration. Dostopno na:  
[http://wwis-dubc1-vip60.adobe.com/products/player\\_census/flashplayer/version\\_penetration.html](http://wwis-dubc1-vip60.adobe.com/products/player_census/flashplayer/version_penetration.html)
- [11] (2010) Microsoft .NET Framework. Dostopno na:  
<http://www.microsoft.com/net/>
- [12] (2010) Microsoft Visual Studio 2008. Dostopno na:  
<http://msdn.microsoft.com/en-us/vstudio/default.aspx>
- [13] (2009) Move window/form without Titlebar in C#. Dostopno na:  
<http://www.codeproject.com/KB/cs/csharpmovewindow.aspx>
- [14] (2010) Multimedijška programska oprema. Dostopno na:  
<http://iris.pfmb.uni-mb.si/old/didgradiva/diplome/sternad/s4.htm>
- [15] (2009) Orodje Articulate – Engage. Dostopno na:  
<http://www.articulate.com/products/studio.php>
- [16] (2009) Orodje Content creator. Dostopno na:  
<http://www.contentgenerator.net/>
- [17] (2010) Orodje iSPRING. Dostopno na:  
<http://www.ispringsolutions.com/>
- [18] (2009) Orodje JClic. Dostopno na:  
<http://clic.xtec.cat/en/jclic/>
- [19] (2009) Orodje Quizfaber. Dostopno na:  
<http://www.lucagalli.net/en/>
- [20] (2009) Orodje Spellmaster. Dostopno na:  
<http://web2tools-mai08.pbworks.com/Spellmaster>

- [21] (2010) Orodje SQL Power Architect. Dostopno na:  
<http://code.google.com/p/power-architect/>
- [22] (2010) Predvajalnik Adobe Flash. Dostopno na:  
<http://www.adobe.com>
- [23] (2010) Recognition of Non-formal and Informal Learning – Home. Dostopno na:  
[http://www.oecd.org/document/25/0,3343,en\\_2649\\_39263238\\_37136921\\_1\\_1\\_1\\_37455,00.html](http://www.oecd.org/document/25/0,3343,en_2649_39263238_37136921_1_1_1_37455,00.html)
- [24] (2009) Spletno okolj Moodle. Dostopno na:  
<http://www.moodle.si>
- [25] (2010) Učbeniški komplet My Sails2. Dostopno na:  
[http://www.zalozba-pivec.com/z\\_ucbeniki0.php](http://www.zalozba-pivec.com/z_ucbeniki0.php)
- [26] (2010) Usage share of operating systems. Dostopno na:  
[http://en.wikipedia.org/wiki/Usage\\_share\\_of\\_operating\\_systems](http://en.wikipedia.org/wiki/Usage_share_of_operating_systems)
- [27] (2010) Visual C#. Dostopno na:  
<http://msdn.microsoft.com/en-us/vcsharp/default.aspx>
- [28] (2010) What is hard interactive whiteboard. Dostopno na:  
<http://www.e4africa.co.za/?p=1438>
- [29] (2010) What is soft interactive whiteboard. Dostopno na:  
<http://www.e4africa.co.za/?p=1443>
- [30] (2010) Windows Forms. Dostopno na:  
<http://msdn.microsoft.com/en-us/library/dd30h2yb.aspx>